

Διατριβή

**ΕΚΤΙΜΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΙΣΜΙΚΟΥ ΜΕ
ΧΡΗΣΗ ΓΕΝΕΤΙΚΟΥ
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ**

Ιάσονος Αγγέλα

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ιανουάριος 2010

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εκτίμηση Κόστους Λογισμικού με χρήση
Γενετικού Προγραμματισμού

Ιάσωνος Αγγέλα

Επιβλέπων Καθηγητής

Ανδρέας Ανδρέου

Η Διατριβή αυτή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του μεταπτυχιακού σε Προηγμένες Τεχνολογίες Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Ιανουάριος 2010

Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα καθηγητή μου, Δρ. Ανδρέου Ανδρέα για την καθοδήγηση και υποστήριξη που μου προσέφερε κατά τη διάρκεια της εκπόνησης της διατριβής μου. Επίσης θα ήθελα να ευχαριστήσω την κυρία Έφη Παπαθεοχάρους για το χρόνο που μου αφιέρωσε, το υλικό και τις γνώσεις που μοιράστηκε μαζί μου σχετικά με την εκτίμηση του κόστους ανάπτυξης λογισμικού και γενικά για τις πολύτιμες συμβουλές που μου έχει δώσει. Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη και την κατανόηση τους.

Περίληψη

Ο ακριβής υπολογισμός της προσπάθειας υλοποίησης ενός λογισμικού αποτελεί ένα πολύ σημαντικό κομμάτι της διαδικασίας ανάπτυξης λογισμικού. Αρχικά αυτός ο υπολογισμός γινόταν μόνο μέσω κάποιων έμπειρων ατόμων όμως αυτό δεν είναι πάντα δυνατό ή όχι και τόσο ακριβές. Επομένως η προσοχή των ερευνητών έχει πλέον στραφεί στην εύρεση ευφυών τρόπων εκμάθησης των ηλεκτρονικών υπολογιστών για να εκτελούν τον υπολογισμό της προσπάθειας ανάπτυξης λογισμικού.

Σκοπός της διπλωματικής αυτής εργασίας είναι με την βοήθεια των γενετικών αλγόριθμων να προσπαθήσει να υπολογίσει το κόστος ανάπτυξης λογισμικού σε κάποια σύνολα δεδομένων με την μεγαλύτερη δυνατή ακρίβεια. Η μελέτη αυτή παρουσιάζει ένα εργαλείο γενετικού προγραμματισμού το οποίο δέχεται ένα αρχείο δεδομένων με ολοκληρωμένα έργα λογισμικού τα οποία χρησιμοποιεί για να μπορεί να καταλήξει στην καλύτερη δυνατή λύση με γνώμονα την προσπάθεια. Τα αρχεία δεδομένων που έχουμε χρησιμοποιήσει είναι αυτά του Cocomo [6] και του Desharnais [15]. Επίσης έχει γίνει προσπάθεια χρήσης ενός μεγάλου συνόλου δεδομένων ISBSG [12], το οποίο αποτελεί διεθνώς αποδεκτό benchmark. Έχουν γίνει κάποιες πειραματικές εκτελέσεις του προτεινόμενου συστήματος και παρουσιάζονται τα αποτελέσματα αυτών των εκτελέσεων, κάποιες γραφικές παραστάσεις καθώς και τα συμπεράσματα που απορρέουν.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή	1
	1.1 Γενική περιγραφή	1
	1.2 Σκοπός εργασίας	2
	1.3 Ανασκόπηση κεφαλαίων	3
Κεφάλαιο 2	Υπολογισμός Κόστους Ανάπτυξης Λογισμικού	4
	2.1 Εισαγωγή	4
	2.2 Γενική περιγραφή	5
	2.3 Παράγοντες που επηρεάζουν την εκτίμηση κόστους	6
	2.4 Μοντέλα υπολογισμού κόστους	7
Κεφάλαιο 3	Γενετικοί Αλγόριθμοι και Γενετικός Προγραμματισμός	11
	3.1 Εισαγωγή	11
	3.2 Ιστορική Αναδρομή	12
	3.3 Περιγραφή Λειτουργίας Γενετικών Αλγορίθμων	13
	3.4 Πλεονεκτήματα - Μειονεκτήματα Γενετικών Αλγορίθμων	16
	3.5 Εφαρμογές Γενετικών Αλγορίθμων	17
	3.6 Γενετικός Προγραμματισμός	18
	3.6.1 Συνάρτηση Καταλληλότητας	20
	3.6.2 Παράγοντας Bloating	20
Κεφάλαιο 4	Προτεινόμενο Σύστημα	22
	4.1 Εισαγωγή	22
	4.2 Εργαλείο GPLAB	22
	4.3 Περιγραφή Προτεινόμενου Συστήματος	25
	4.3.1 Επιλογή Αρχείου Δεδομένων	27
	4.3.2 Επιλογή Παραμέτρων	29
	4.3.3 Εκτέλεση Γενετικού Προγραμματισμού	36
	4.3.4 Δημιουργία Αρχείου Εξόδου	39
	4.3.5 Παραγόμενες Γραφικές Παραστάσεις	44
Κεφάλαιο 5	Πειραματική Διαδικασία	48
	5.1 Εισαγωγή	48
	5.2 Γενική Ανασκόπηση Πειραμάτων	49
	5.2.1 Πειραματικές Εκτελέσεις Αριθμητικών Συναρτήσεων	50
	5.2.2 Πειραματικές Εκτελέσεις Λογικών Συναρτήσεων	53
	5.2.3 Πειραματικές Εκτελέσεις Κατηγορικών Συναρτήσεων	55
	5.3 Παρουσίαση Αποτελεσμάτων	56
	5.3.1 Αποτελέσματα Αριθμητικών Εκτελέσεων	56
	5.3.2 Αποτελέσματα Λογικών Εκτελέσεων	59
	5.3.3 Αποτελέσματα Κατηγορικών Εκτελέσεων	60
	5.4 Συμπεράσματα και Γραφικές Παραστάσεις	60
Κεφάλαιο 6	Συμπεράσματα και Εισηγήσεις για Μελλοντική Εργασία	72

6.1	Συμπεράσματα	72
6.2	Εισηγήσεις για μελλοντική εργασία	73
Βιβλιογραφία	76
Παράρτημα Α	Παρουσίαση Αποτελεσμάτων Εκτελέσεων	78
A.1	Εισαγωγή	78
A.2	Αποτελέσματα Αριθμητικών Εκτελέσεων – Αρχείο Cocomo	79
A.3	Αποτελέσματα Αριθμητικών Εκτελέσεων – Αρχείο Desharnais	94
A.4	Αποτελέσματα Λογικών Εκτελέσεων – Αρχείο Cocomo	105
A.5	Αποτελέσματα Λογικών Εκτελέσεων – Αρχείο Desharnais	110
A.6	Αποτελέσματα Κατηγορικών Εκτελέσεων – Αρχείο ISBSG	114
Παράρτημα Β	Γραφικές Παραστάσεις Καλύτερων Εκτελέσεων	117
B.1	Εισαγωγή	117
B.2	Γραφικές Παραστάσεις Καλύτερων Εκτελέσεων	117
B.2.1	Παραστάσεις Αριθμητικών Εκτελέσεων Cocomo	118
B.2.2	Παραστάσεις Αριθμητικών Εκτελέσεων Desharnais	133
B.2.3	Παραστάσεις Λογικών Εκτελέσεων Cocomo	142
B.2.4	Παραστάσεις Λογικών Εκτελέσεων Desharnais	147
B.2.5	Παραστάσεις Κατηγορικών Εκτελέσεων ISBSG	151

Κεφάλαιο 1

Εισαγωγή

- 1.1 Γενική Περιγραφή
 - 1.2 Σκοπός
 - 1.3 Ανασκόπηση Κεφαλαίων
-

1.1 Γενική Περιγραφή

Ένα από τα μεγαλύτερα προβλήματα που ερχόμαστε αντιμέτωποι σε έργα ανάπτυξης πληροφοριακών συστημάτων είναι ο υπολογισμός του κόστους ανάπτυξης ενός καινούριου λογισμικού συστήματος. Ο υπολογισμός αυτός είναι μια δύσκολη αλλά και επίπονη διαδικασία η οποία θα πρέπει να γίνει στα πρώτα στάδια του κύκλου ανάπτυξης της ζωής ενός λογισμικού συστήματος. Η σημαντικότητα της σωστής και έγκαιρης πρόβλεψης του κόστους ανάπτυξης ενός λογισμικού είναι μεγάλη αφού έχει άμεση επίδραση στην λήψη των σωστών αποφάσεων στη συνέχεια του έργου από όλα τα εμπλεκόμενα μέρη. Το υπολογιζόμενο κόστος ενός υπό υλοποίηση λογισμικού όσον αφορά τόσο το χρηματικό μέρος (budget) αλλά και το χρονικό μέρος (schedule), μπορεί να επηρεάσει άμεσα την λήψη αποφάσεων από τους διοικούντες. Πολλές ήταν οι περιπτώσεις όπου λογισμικά είχαν σταματήσει πριν καν ξεκινήσουν αφού το υπολογιζόμενο κόστος σε χρόνο ή σε χρήμα ήταν αποτρεπτικά.

Προσπάθειες για εύρεση βέλτιστων τρόπων υπολογισμού του κόστους ανάπτυξης λογισμικού έχουν γίνει περίπου από τις δεκαετίες του 1960 και συνεχίζονται μέχρι σήμερα. Πολλά είναι τα μοντέλα που έχουν προταθεί από διάφορους τομείς της πληροφορικής όπως είναι τα νευρωνικά δίκτυα ή οι γενετικοί αλγόριθμοι, χωρίς ωστόσο κάποιο από αυτά να λύνει το μεγάλο αυτό πρόβλημα. Η δυσκολία υπολογισμού του κόστους επηρεάζεται από διάφορους παράγοντες όπως είναι η αβεβαιότητα των έργων στο κόστος, ο καταμερισμός των πόρων και το χρονοδιάγραμμα. Ο υπολογισμός του

κόστους επηρεάζεται και από τις διάφορες αλλαγές στην τεχνολογία, στο προσωπικό ή ακόμα και στις απαιτήσεις των τελικών χρηστών που μπορεί να υπάρξουν. Συνεπώς διαφαίνεται ότι πολλοί είναι οι αστάθμητοι παράγοντες που μπορεί να επηρεάσουν και να καθορίσουν την επιτυχημένη ή αποτυχημένη κατάληξη του υπό υλοποίηση λογισμικού συστήματος.

1.2 Σκοπός

Σκοπός της μελέτης αυτής είναι η εκτίμηση της προσπάθειας που χρειάζεται να καταβληθεί στην υλοποίηση ενός λογισμικού. Πέρα όμως από αυτό, στόχος αποτελεί και η εξαγωγή κανόνων οι οποίοι θα μπορούν να φανούν χρήσιμοι στην διαδικασία εκτίμησης του κόστους ανάπτυξης λογισμικού. Μέσα από αυτούς τους κανόνες θα προσπαθήσουμε επίσης να βρούμε ποιοι πιθανόν να είναι οι κρίσιμοι παράμετροι, δηλαδή οι παράμετροι που είναι πολύ σημαντικοί στην εκτίμηση του κόστους.

Το μοντέλο που θα υλοποιήσουμε στην παρούσα μελέτη, βασίζεται σε γενετικούς αλγόριθμους, δέχεται σαν είσοδο ένα αρχείο μορφής excel, με δεδομένα άλλων υλοποιημένων λογισμικών και περνώντας μέσα από το στάδιο της εκμάθησης να παράξει κανόνες ευρέως αποδεκτούς με βάση τα δεδομένα. Η βασική μετρική σε αυτή την διαδικασία είναι ο υπολογισμός της προσπάθειας που χρειάζεται να καταβληθεί προκειμένου να υλοποιηθεί ένα έργο. Οι γενετικοί αλγόριθμοι θα μας βοηθήσουν στο να βρούμε την καλύτερη δυνατή (βέλτιστη) λύση αφού όπως είναι γνωστό μπορούν να δουλέψουν και να παράξουν το καλύτερο δυνατό αποτέλεσμα σε πολυδιάστατα περιβάλλοντα. Θα γίνει μια σειρά από πειράματα και κατά συνέπεια θα παρουσιαστούν μια σειρά από αποτελέσματα. Αυτά τα αποτελέσματα θα μπορέσουμε να τα αναλύσουμε μέσω κάποιων συντελεστών λάθους και να δούμε ποια είναι η πραγματική αξία του κάθε ενός κανόνα με βάση την τιμή καταλληλότητας που θα έχουν.

Κάποιος μπορεί να διερωτηθεί γιατί έχουμε κάνει χρήση γενετικών αλγορίθμων και όχι κάποια άλλης τεχνικής. Η απάντηση είναι επειδή οι γενετικοί αλγόριθμοι αποτελούν ένα από τους καλύτερους τρόπους επίλυσης προβλημάτων για τα οποία δεν γνωρίζουμε πολλές πληροφορίες [25]. Λόγω της γενικότητας τους μπορούν να αποδώσουν καλά σε

οποιοδήποτε χώρο. Επίσης θέλαμε με αυτή τη μελέτη να ερευνήσουμε πως δουλεύουν και αντιδρούν οι γενετικοί αλγόριθμοι πάνω σε λογικά και κατηγορικά δεδομένα.

1.3 Ανασκόπηση Κεφαλαίων

Στο κεφάλαιο ένα γίνεται μια μικρή εισαγωγή για τον υπολογισμό του κόστους ανάπτυξης λογισμικού καθώς επίσης και για το στόχο της παρούσας μελέτης.

Στο δεύτερο κεφάλαιο γίνεται μια πιο λεπτομερής ανάλυση της θεωρίας του υπολογισμού του κόστους ανάπτυξης λογισμικού, αναφέρονται οι παράγοντες που το επηρεάζουν καθώς επίσης και τα διάφορα μοντέλα υπολογισμού του κόστους.

Στο τρίτο κεφάλαιο γίνεται μια ιστορική αναδρομή αλλά και μια περιγραφή της λειτουργίας των γενετικών αλγορίθμων, αναλύονται να πλεονεκτήματα αλλά και τα μειονεκτήματα των γενετικών αλγορίθμων, δίνονται παραδείγματα εφαρμογών τους και τέλος γίνεται μια αναφορά στο γενετικό προγραμματισμό.

Το κεφάλαιο τέσσερα ασχολείται με την επεξήγηση του προτεινόμενου συστήματος. Δίνεται μια σύντομη περιγραφή του εργαλείου GPLab ενώ στη συνέχεια γίνεται μια πιο λεπτομερής περιγραφή του προτεινόμενου συστήματος. Περιγράφονται οι είσοδοι του συστήματος, τα κριτήρια αξιολόγησης αλλά και οι παραγόμενες γραφικές παραστάσεις.

Το πέμπτο κεφάλαιο έχει να κάνει ολοκληρωτικά με την πειραματική μας διαδικασία. Περιγράφεται η μεθοδολογία που ακολουθήθηκε, ο σχεδιασμός των διάφορων πειραμάτων που έγιναν και παρουσιάζονται τα αποτελέσματα που έχουμε πάρει καθώς και τα συμπεράσματα που βγαίνουν μέσα από αυτά.

Τέλος το κεφάλαιο έξι παρέχει μια συνοπτική περιγραφή των συμπερασμάτων μέσα από την πειραματική διαδικασία καθώς επίσης και κάποιες εισηγήσεις για μελλοντικές επεκτάσεις αυτής της μεθόδου.

Κεφάλαιο 2

Υπολογισμός Κόστους Ανάπτυξης Λογισμικού

- 2.1 Εισαγωγή
 - 2.2 Γενική Περιγραφή
 - 2.3 Παράγοντες που Επηρεάζουν την Εκτίμηση Κόστους
 - 2.4 Μοντέλα Υπολογισμού Κόστους
-

2.1 Εισαγωγή

Η ανάπτυξη λογισμικού στην εποχή μας, είναι η πιο αναγκαία, πολύτιμη αλλά και η πιο ακριβής διαδικασία των έργων ανάπτυξης πληροφοριακών συστημάτων. Αυτό είναι αποτέλεσμα του τεχνολογικού περιβάλλοντος που ζούμε και το οποίο απαιτεί συνεχώς πιο αυτοματοποιημένες μεθόδους εργασίας. Για να δημιουργηθεί ένα λογισμικό πρέπει να ληφθούν υπόψη και να αναλυθούν πολλοί παράμετροι έτσι ώστε να μπορέσουμε να έχουμε το επιθυμητό αποτέλεσμα τόσο χρονικά αλλά και ποιοτικά. Συνεπώς υπάρχει ανάγκη πρόβλεψης κάποιων βασικών χαρακτηριστικών που διέπουν το κόστος ανάπτυξης λογισμικού στα αρχικά στάδια της διαδικασίας ανάπτυξης λογισμικού. Ένα από τα πιο βασικά χαρακτηριστικά είναι ο υπολογισμός του κόστους αλλά και του όγκου εργασίας που χρειάζεται μια συγκεκριμένη εφαρμογή προκειμένου να υλοποιηθεί. Άμεσα συνδεδεμένα με το κόστος και τον όγκο εργασίας είναι επίσης και ο υπολογισμός των απαιτούμενων πόρων σε ανθρώπινο δυναμικό, η επιλογή και διαχείριση της ανάπτυξης συστημάτων λογισμικού, ο χρονικός προγραμματισμός των εργασιών αλλά και η παρακολούθηση και ο έλεγχος ποιοτικών χαρακτηριστικών του έργου και του συστήματος που αναπτύσσεται. Για τους σκοπούς της παρούσας μελέτης εμείς επιλέξαμε να ασχοληθούμε με το πρώτο κομμάτι που έχουμε αναφέρει πιο πάνω (Cost & Effort Estimation).

2.2 Γενική Περιγραφή

Η έννοια του κόστους ανάπτυξης λογισμικού δεν αφορά ένα και μόνο παράγοντα αλλά είναι μια ευρύτερη έννοια η οποία αναφέρεται σε διάφορες πτυχές ενός λογισμικού [2]. Μπορεί να αναφέρεται σε κόστος της προσπάθειας που καταβάλλεται (effort), στη διάρκεια του κύκλου ανάπτυξης του έργου (duration), ή σε κόστος χρημάτων (money). Το πιο συνηθισμένο κόστος είναι αυτό της προσπάθειας το οποίο μετρείται με άνθρωπο-ώρες και πιο συγκεκριμένα σε άνθρωπο-μήνες (person-months). Έχει αποδειχτεί διεθνώς ότι η ανάπτυξη ενός λογισμικού τις περισσότερες φορές (αν όχι και πάντα) υπερβαίνει τον αρχικό προϋπολογισμό αλλά επίσης επεκτείνεται και σε χρόνο ανάπτυξης λόγω ανεπαρκών αρχικών εκτιμήσεων. Η βιωσιμότητα πολλών οργανισμών που αναλαμβάνουν να διεκπεραιώσουν τέτοια έργα ανάπτυξης λογισμικών είναι άμεσα συνδεδεμένη με την παραγωγή του λογισμικού τόσο εντός χρονικών ορίων αλλά και εντός χρηματικών πλαισίων. Είναι πολλά τα παραδείγματα που τέτοιοι οργανισμοί δεν επέζησαν αφού η μη διεκπεραίωση έργων τους μέσα στα συμφωνηθέντα πλαίσια δεν έγινε εφικτή λόγω αυτής της μη σωστής κοστολόγησης πράγμα που είχε αντίκτυπο στη φήμη, την απόδοση αλλά και την ανταγωνιστικότητα των εν λόγω εταιρειών. Με βάση τα πιο πάνω, είναι λογικό η ανάγκη για εύρεση τρόπων που να μπορούν να κοστολογούν σωστά ένα λογισμικό πριν ξεκινήσει ακόμα ο κύκλος ζωής του είναι άμεση αλλά και πάρα πολύ σημαντική εφόσον ένα τέτοιο μοντέλο θα αποφέρει εξοικονόμηση σε χρήματα που είναι ο υπ' αριθμόν ένα παράγοντας σε κάθε βήμα της ζωής μας.

Η υπολογισμός του κόστους σε κάθε λογισμικό πρέπει να βασίζεται στη φύση της εφαρμογής κατά πρώτιστο λόγο. Πρέπει να λαμβάνονται υπόψη το μέγεθος, η πολυπλοκότητα, το περιβάλλον ανάπτυξης, η γλώσσα προγραμματισμού, το προσωπικό που θα ασχοληθεί με την ανάπτυξη του λογισμικού αλλά και οι απαιτήσεις των τελικών χρηστών [2]. Η εκτίμηση του κόστους αυτού θα παίξει ιδιαίτερο ρόλο στην διαπραγμάτευση των συμβολαίων μεταξύ πελάτη και εταιρείας λογισμικού, στο συντονισμό και έλεγχο των διαδικασιών αλλά επίσης θα βοηθήσει και σε μελλοντικές προβλέψεις παρόμοιων εφαρμογών. Είναι κατανοητό ότι το κόστος λογισμικού δεν αφορά μόνο τους διαχειριστές της εταιρείας ανάπτυξης αλλά και όλους τους άμεσα ή

έμμεσα εμπλεκόμενους φορείς: ιδιοκτήτες εταιρείας, τελικούς χρήστες, δημιουργούς του συστήματος.

Όπως διαφαίνεται και από τα πιο πάνω, μια τέτοια κοστολόγηση ενός έργου πρέπει να γίνει με όσο το δυνατό πιο ακριβής μεθόδους έτσι ώστε να αποφύγουμε φαινόμενα όπως είναι η υπερεκτιμήσεις (overestimate) ή οι υποεκτιμήσεις (underestimate) του κόστους. Οι περιπτώσεις των υπερεκτιμήσεων συνήθως έχουν ως αρνητικό αποτέλεσμα να μην υπογράφεται τελικά η συμφωνία μεταξύ των δυο πλευρών εφόσον το κόστος παρουσιάζεται πολύ ψηλό. Επίσης μπορεί με μια τέτοια περίπτωση να γίνουν λανθασμένες κατανομές σε πόρους και η παραγωγικότητα των προγραμματιστών αλλά και όλων των εμπλεκόμενων να είναι χαμηλή. Από την άλλη τυχόν υποεκτίμηση του κόστους μπορεί να επιφέρει έργα τα οποία να υπερβούν κατά πολύ το αρχικό τους κόστος, η κατανομή των διάφορων διαθέσιμων πόρων να γίνει λανθασμένα ή ακόμα και η ποιότητα του προϊόντος να μην είναι η αναμενόμενη αλλά πιο χαμηλή. Υποεκτίμηση μπορεί να γίνει και σε χρόνο οπότε σε τέτοιες περιπτώσεις θα παρατηρηθεί καθυστέρηση στην παράδοση του τελικού προϊόντος. Ένας ακριβής υπολογισμός της προσπάθειας (effort) θα βοηθήσει ιδιαίτερα την διαδικασία έκδοσης προτεραιοτήτων στα διάφορα έργα και με αυτό τον τρόπο να γίνει καλύτερος καταμερισμός εργασιών στα διάφορα εμπλεκόμενα άτομα αλλά και να γίνεται έλεγχος της όλης διαδικασίας ανάπτυξης λογισμικού.

2.3 Παράγοντες που Επηρεάζουν την Εκτίμηση Κόστους

Προκειμένου να υπολογίσουμε το κόστος ανάπτυξης λογισμικού πρέπει να κάνουμε πρόβλεψη της προσπάθειας (effort) που χρειάζεται προκειμένου να έρθει εις πέρας το όλο έργο. Η πρόβλεψη μπορεί να αποβεί χρήσιμη στα πρώτα στάδια αλλά και σε ολόκληρο τον κύκλο ζωής ενός λογισμικού. Το ιδανικότερο είναι η πρόβλεψη του κόστους να προσεγγίσει όσο το δυνατό περισσότερο τις πραγματικές τιμές. Με αυτό τον τρόπο θα μπορέσουμε να λάβουμε αποφάσεις οι οποίες να είναι οι όσο το δυνατό καλύτερες και το έργο να διεκπεραιωθεί μέσα σε όλα τα πλαίσια που έχουμε θέσει.

Μπορεί κανείς εύκολα να αντιληφθεί ότι η διαδικασία ανάπτυξης λογισμικού είναι πολύπλοκη αλλά συνάμα και μοναδική εφόσον ποτέ κανένα λογισμικό σύστημα δεν είναι ακριβώς το ίδιο με κάποιο άλλο. Η κυριότερη δυσκολία στην εκτίμηση του κόστους είναι αποτελεί η έλλειψη εκπαιδευμένων ατόμων με εμπειρία που να μπορούν να κάνουν τέτοιες εκτιμήσεις. Υπάρχουν όμως πέρα από αυτό το λόγο και διάφοροι άλλοι λόγοι που πολλές φορές κάνουν δύσκολή ίσως και αδύνατη την εκτίμηση του κόστους ενός έργου λογισμικού. Τέτοιο παράγοντες είναι και οι ακόλουθοι:

- Η αβεβαιότητα των διαθέσιμων δεδομένων κατά την έναρξη του έργου.
- Η πολυπλοκότητα του συστήματος που πρόκειται να αναπτυχθεί.
- Μέγεθος του υπό υλοποίηση συστήματος.
- Ικανότητες των ατόμων της ομάδας ανάπτυξης λογισμικού.
- Αριθμός ατόμων της ομάδας ανάπτυξης λογισμικού.
- Εργαλεία που μπορούν να έχουν στην διάθεση τους τα άτομα της ομάδας ανάπτυξης λογισμικού.
- Εμπειρία στην ανάπτυξη παρόμοιων εφαρμογών.
- Συχνές αλλαγές στις απαιτήσεις του πελάτη.
- Ανάγκη συμμόρφωσης με διάφορα πρότυπα.
- Έλλειψη δεδομένων παλιών εφαρμογών σε αριθμητικό κόστος.
- Δεν υπάρχουν ομογενή και αξιόπιστα δεδομένα, δηλαδή πληροφορίες από έργα που έγιναν σε ίδια περιβάλλοντα εργασίας.

2.4 Μοντέλα Υπολογισμού Κόστους

Όπως έχουμε δει και πιο πάνω υπάρχουν πολλοί αστάθμητοι παράγοντες που δεν επιτρέπουν τον εύκολο υπολογισμό του κόστους. Ο υπολογισμός του κόστους λογισμικού ασχολείται με την πρόβλεψη των πόρων που είναι απαραίτητοι για την

ανάπτυξη αλλά και την συντήρηση ενός λογισμικού συστήματος. Υπάρχει ένα ευρύ φάσμα από μοντέλα υπολογισμού του κόστους λογισμικού με τα οποία έχουν γίνει προσπάθειες να γίνει όσο το δυνατό καλύτερη εκτίμηση του κόστους χωρίς όμως οι περισσότερες να είναι επιτυχείς. Τα μοντέλα υπολογισμού του κόστους μπορεί να διαχωριστούν σε δύο μεγάλες κατηγορίες: την κατηγορία βασιζόμενη σε μοντέλα (model-based) και την κατηγορία μη βασιζόμενη σε μοντέλα (non model-based).

Η κατηγορία του υπολογισμού που βασίζεται σε μοντέλα αποτελείται από μια μέθοδο μοντελοποίησης, ένα μοντέλο αλλά και μια μέθοδο εφαρμογής του μοντέλου αυτού. Το μοντέλο παίρνει ένα σύνολο από εισόδους ενώ το αποτέλεσμα του θα είναι το effort του έργου. Από την άλλη η κατηγορία του υπολογισμού που δεν βασίζεται σε μοντέλα μπορεί να συνδυάσει μια ή και περισσότερες τεχνικές με την βοήθεια των οποίων θα παράξουν ένα υπολογισμό του κόστους. Σε αυτή την κατηγορία είναι απαραίτητη η συμμετοχή expert έτσι ώστε το παραγόμενο κόστος να είναι σχετικά αντικειμενικό.

Κάποιες τεχνικές που υπάγονται στις πιο πάνω δύο κατηγορίες και που έχουν ακολουθηθεί κατά διαστήματα για να γίνει εκτίμηση του κόστους είναι οι ακόλουθες [18]:

- Εκτίμηση Top-Down: Για να μπορέσουμε να κάνουμε εκτίμηση με την μέθοδο αυτή, ξεκινάμε από το σύστημα σαν σύνολο και προσδιορίζουμε τα επιμέρους υποσυστήματα που πρέπει να υλοποιηθούν, και υπολογίζουμε τα επί μέρους κόστη. Η διαδικασία συνεχίζεται μέχρι τις απλές μονάδες που δεν απαιτούν περαιτέρω διάσπαση. Μπορεί να λαμβάνεται υπόψη και το κόστος της επί μέρους ένωσης των υποσυστημάτων, όμως μπορεί να υπο-εκτιμήσουμε το κόστος της ανάπτυξης των τελικών μονάδων της εφαρμογής.
- Εκτίμηση Bottom-Up: Αυτή η μέθοδος είναι η αντίθετη της προηγούμενης αφού εδώ ξεκινάμε από τις επιμέρους μονάδες του συστήματος και το κόστος ανάπτυξης της κάθε μονάδας υπολογίζεται ανεξάρτητα και αθροίζεται με το κόστος των άλλων μονάδων, για να μας δώσει το τελικό κόστος ανάπτυξης της εφαρμογής. Αυτή η μέθοδος είναι ακριβής και ορθή φτάνει να έχουμε κάνει μια πολύ καλή σχεδίαση του συστήματος. Όμως και πάλι εδώ μπορεί να γίνει υπό-

εκτίμηση του κόστους ανάπτυξης εάν υπό-εκτιμήσουμε το κόστος ενοποίησης και τεκμηρίωσης.

- Εκτίμηση Function Point: Με την μέθοδο αυτή υπολογίζονται τα function points (FPs), γίνεται εκτίμηση της πολυπλοκότητας της εφαρμογής και γίνεται χρήση εμπειρικής αναλογίας σχέσης της προσαρμοσμένης μετρικής FP με lines of code (LOC) και ανθρωπόωρες για τη συγκεκριμένη γλώσσα προγραμματισμού που χρησιμοποιείται για τη ανάπτυξη της εφαρμογής.
- Εκτίμηση κατοχύρωσης έργου - Pricing to win: Το κόστος του έργου είναι ουσιαστικά αυτό που θα πληρώσει ο πελάτης. Με αυτό τον τρόπο μπορεί μεν να κατοχυρώνουμε το έργο εντούτοις όμως η πιθανότητα ο πελάτης να παραλάβει τελικά την εφαρμογή έτσι όπως θα την ήθελε είναι χαμηλή. Επίσης με την μέθοδο αυτή το κόστος ανάπτυξης δεν αντικατοπτρίζει σωστά την εργασία που απαιτείται για την εκτέλεση του έργου.
- Εκτίμηση κατά αναλογία – Estimation by analogy: Το κόστος της εφαρμογής εκτιμάται κατά αναλογία με το κόστος άλλων γνωστών και παρόμοιων εφαρμογών. Ένα σημαντικό θέμα με αυτή τη μεθοδολογία είναι η ρύθμιση των διάφορων παραμέτρων (calibration) στα τοπικά δεδομένα της δικής μας εφαρμογής. Αυτή η μεθοδολογία είναι εύκολη στην εφαρμογή, ακριβής και ορθή μόνο όμως εάν υπάρχουν ιστορικά δεδομένα για την κατά αναλογία εκτίμηση του κόστους, ενώ δεν είναι εφαρμόσιμη εάν δεν έχουμε στοιχεία. Επομένως για να μπορεί να χρησιμοποιηθεί αυτή η μέθοδος θα πρέπει να γίνετε συστηματική καταγραφή δεδομένων στοιχείων κόστους για κάθε έργο που έχει τελειώσει. Επίσης εδώ παρουσιάζονται και δυσκολίες στην θεωρητική μελέτη του σφάλματος πρόβλεψης (prediction error).
- Εκτίμηση με βάση τους έμπειρους - Expert Judgment: Ένας ή περισσότεροι εμπειρογνώμονες στις περιοχές πεδίου εφαρμογής και στις περιοχές ανάπτυξης συστημάτων εκτιμούν το κόστος ανάπτυξης. Η διαδικασία επαναλαμβάνεται όσες φορές χρειάζεται μέχρι που να βρεθεί μια κοινή γνώμη και όλοι οι εμπειρογνώμονες είναι ικανοποιημένοι από το αποτέλεσμα. Κάθε εμπειρογνώμονας εκφράζει και μια διαφορετική σκοπιά για την τελική εκτίμηση

του κόστους. Αυτή η μέθοδος είναι σχετικά φτηνή σε κόστος όμως δεν είναι πολύ αξιόπιστη αφού είναι δύσκολο να βρούμε καλούς εμπειρογνώμονες. Όμως αποτελεί την μόνη λύση σε περιπτώσεις που είτε υπάρχει εφαρμογή μιας νέας τεχνολογίας είτε δεν έχουμε καθόλου ιστορικά δεδομένα από μετρήσεις άλλων ολοκληρωμένων εφαρμογών.

- Εκτίμηση με το νόμο του Parkinson: Με βάση αυτό το νόμο το έργο θα κοστίσει όσο κοστίζουν οι διαθέσιμοι πόροι. Το μεγάλο πλεονέκτημα αυτής της μεθόδου είναι ότι θα είμαστε εντός του προϋπολογισμού μας, αλλά από την άλλη η εφαρμογή δεν παραδίδεται ή παραδίδεται αλλά όχι τελειωμένη.
- Εκτίμηση με αλγοριθμικά μοντέλα: Το κόστος υπολογίζεται με ένα αλγοριθμικό τρόπο σαν συνάρτηση του τύπου της εφαρμογής. Σε αυτή την τεχνική γίνεται συνδυασμός κάποιων εξισώσεων, προσαρμογής δεδομένων αλλά και εξειδικευμένης γνώσης.

Ανεξάρτητα από το ποια μεθοδολογία χρησιμοποιείται, ο τομέας της βιομηχανίας έχει επισημάνει ότι παρουσιάζονται ιδιαίτερα προβλήματα στον καταμερισμό των πόρων αλλά και στον χρονοπρογραμματισμό και επομένως είναι άμεση η ανάγκη για βελτίωση αυτών των μεθοδολογιών εκτίμησης του κόστους.

Κεφάλαιο 3

Γενετικοί Αλγόριθμοι και Γενετικός Προγραμματισμός

- 3.1 Εισαγωγή
 - 3.2 Ιστορική Αναδρομή
 - 3.3 Περιγραφή Λειτουργίας Γενετικών Αλγορίθμων
 - 3.4 Πλεονεκτήματα – Μειονεκτήματα Γενετικών Αλγορίθμων
 - 3.5 Εφαρμογές Γενετικών Αλγορίθμων
 - 3.6 Γενετικός Προγραμματισμός
-

3.1 Εισαγωγή

Οι Γενετικοί αλγόριθμοι είναι μια διαδικασία προσομοίωσης της φύσης σε ότι αφορά την εξέλιξη των ειδών η οποία έχει αναπτυχθεί τα τελευταία χρόνια προκειμένου να αντιμετωπισθούν προβλήματα επίλυσης συστημάτων βασισμένων στις αρχές της αποτίμησης αλλά και της κληρονομικότητας. Η χρήση των αλγορίθμων αυτών για την επίλυση δύσκολων προβλημάτων τα τελευταία χρόνια έχει αυξηθεί αφού έχει αποδειχτεί ότι οι γενετικοί αλγόριθμοι είναι χρήσιμοι στην αναζήτηση λύσεων προβλημάτων βελτιστοποίησης και εκμάθησης. Με τους γενετικούς αλγορίθμους μπορούμε να μοντελοποιήσουμε δεδομένα με αποτελεσματικό τρόπο και να παράξουμε μαθηματικές εξισώσεις που περιγράφουν με επιτυχία τις διαδικασίες μας.

Οι γενετικοί αλγόριθμοι είναι μια τεχνική στοχαστικής αναζήτησης και στηρίζονται στην αρχή του Δαρβίνου στην οποία μέσα από ένα πληθυσμό εμβρύων όντων επιζούν μόνο τα υγιέστερα αφού με βάση τον νόμο των πιθανοτήτων μόνο αυτά μπορούν να αντεπεξέλθουν στις δυσκολίες του περιβάλλοντος στο οποίο ανήκουν.

3.2 Ιστορική Αναδρομή

Η πρώτη φορά που έκαναν την εμφάνιση τους οι γενετικοί αλγόριθμοι ήταν στα τέλη της δεκαετίας του 1950 με αρχές της δεκαετίας του 1960 από μια ομάδα βιολόγων οι οποίοι έψαχναν τρόπους μοντελοποίησης των πτυχών της φυσικής εξέλιξης [21]. Γύρω στο 1962 μερικοί ερευνητές κατάφεραν να αναπτύξουν αλγόριθμους εμπνευσμένους από τις έννοιες της εξέλιξης για βελτιστοποίηση λειτουργιών και εκμάθηση μηχανών χωρίς όμως να καταφέρουν περισσότερη έρευνα στον τομέα αυτό. Αργότερα γύρω στο 1965 γίνεται μια πιο προσεκτική επέκταση στον τομέα αυτό, όταν ο Ingo Rechenberg εισήγαγε την τεχνική της εξελικτικής στρατηγικής. Σε αυτή την προσπάθεια δεν υπήρχαν οι έννοιες του πληθυσμού ή των γενετικών τελεστών, αλλά κάθε γονέας παρήγαγε ένα απόγονο μέσω μετάλλαξης και ο καλύτερος από αυτούς τους δύο επικρατούσε και παρέμενε για την επόμενη μετάλλαξη. Η έννοια του πληθυσμού είχε ενταχτεί αργότερα σε μεταγενέστερες προσπάθειες.

Η αρχή του εξελικτικού προγραμματισμού ήρθε αργότερα γύρω στο 1966 από τους L.J. Fogel, A.J. Owens και M.J. Walsh. Οι ερευνητές αυτοί εισήγαγαν την έννοια των υποψήφιων λύσεων και τις αναπαριστούσαν με απλές και πεπερασμένες καταστάσεις μηχανών από τις οποίες επιλεγόταν μια με τυχαίο τρόπο για να μεταλλαχτεί και η καλύτερη από τις δύο να επικρατήσει. Και σε αυτήν την προσπάθεια όμως δεν υπάρχει και πάλι η έννοια των γενετικών τελεστών.

Το 1975, ο John Holland με την βοήθεια κάποιων συνεργατών του εισήγαγε τις έννοιες των προσαρμοστικών ψηφιακών συστημάτων μέσω μετάλλαξης, επιλογής και διασταύρωσης, προσομοιώνοντας με αυτό τον τρόπο την διαδικασία της βιολογικής εξέλιξης σαν μια στρατηγική επίλυσης προβλημάτων. Παράλληλα με αυτό, ο Kenneth De Jong μπόρεσε να δείξει τις μεγάλες δυνατότητες των γενετικών αλγορίθμων ακόμα και κάτω από θορυβώδες, μη συνεχόμενες περιοχές αναζήτησης.

Με βάση αυτές τις αρχές, παρατηρείται στις αρχές με μέσα της δεκαετίας του 1980 χρήση των γενετικών αλγορίθμων σε ένα ευρύ φάσμα αντικειμένων από αφηρημένα μαθηματικά προβλήματα μέχρι μηχανολογικά θέματα όπως είναι η αναγνώριση προτύπων και η κατηγοριοποίηση.

Φτάνοντας στο σήμερα, βλέπουμε ότι η δύναμη της εξέλιξης έχει αγγίξει σχεδόν όλους τους τομείς και νέες χρήσεις των γενετικών αλγόριθμων ανακαλύπτονται συνεχώς καθώς η έρευνα στον τομέα αυτό συνεχίζεται.

3.3 Περιγραφή Λειτουργίας Γενετικών Αλγορίθμων

Οι γενετικοί αλγόριθμοι αποτελούν μια μέθοδο αναζήτησης βέλτιστων λύσεων σε πολυδιάστατα προβλήματα στα οποία δεν υπάρχει τρόπος να βρούμε τον ιδανικότερο συνδυασμό μεταξύ μεταβλητών και τιμών έτσι ώστε να έχουμε την βέλτιστη δυνατή λύση [28]. Οι γενετικοί αλγόριθμοι έχουν σαν βάση τους την επιστήμη της βιολογίας και βασίζονται στην έννοια της εξέλιξης. Ένα από τα βασικότερα τους χαρακτηριστικά είναι και το γεγονός ότι προσπαθούν να μιμηθούν την διαδικασία της βιολογικής εξέλιξης. Βασικό τους στοιχείο επίσης αποτελούν και οι επαναλήψεις μέσα από τις οποίες ο αλγόριθμος μας θα περάσει από το στάδιο της εκμάθησης για να δώσει στο τέλος το καλύτερο επιθυμητό αποτέλεσμα.

Αρχικά δημιουργείται ένας τυχαίος συνδυασμός των μεταβλητών δημιουργώντας ένα συνδυασμό από λύσεις. Η κάθε λύση στην συνέχεια θα εκτιμηθεί μέσω μιας συνάρτησης. Αυτή η συνάρτηση θα μας καθορίσει ποιες από τις λύσεις μας είναι πιο κοντά στην επιθυμητή και με βάση αυτές θα προχωρήσουμε στην μετάλλαξη τους για να δημιουργήσουμε την επόμενη γενιά. Μέσω των επαναλήψεων θα επιβιώσουν μόνο οι πιο δυνατές οι οποίες θα αποτελέσουν και τις μαθηματικές εξισώσεις του αποτελέσματος.

Οι γενετικοί αλγόριθμοι βασίζονται στην θεωρία της εξέλιξης. Σε αυτή την θεωρία οι οργανισμοί που δεν μπορούν να επιβιώσουν στο περιβάλλον τους πεθαίνουν, ενώ οι υπόλοιποι θα πολλαπλασιαστούν μέσω της αναπαραγωγής. Οι απόγονοι κάθε γενιάς παρουσιάζουν κάποιες διαφοροποιήσεις από τους προγόνους τους και υπερσχύουν μόνο οι απόγονοι με τα δυνατότερα χαρακτηριστικά.

Στον πιο κάτω πίνακα μπορούμε να δούμε την αντιστοιχία της ορολογίας της επιστήμης της βιολογίας με τους γενετικούς αλγόριθμους:

Βιολογικό πρότυπο	Γενετικοί αλγόριθμοι
Γονίδιο	Παράμετρος (μεταβλητή)
Χρωμόσωμα	Υποψήφια λύση (συμβολοσειρά)
Πληθυσμός	Σύνολο υποψήφιας λύσεων
Πιθανότητα επιβίωσης	Συνάρτηση βελτιστοποίησης
Μετάλλαξη	Τυχαία αναζήτηση λύσης
Διασταύρωση	Σύνθεση λύσεων
Γενιά	Ανακύκλωση

Πίνακας 3.1: Βιολογικό Πρότυπο και Γενετικοί Αλγόριθμοι [23]

Όπως έχουμε ήδη αναφέρει, ένα από τα βασικά χαρακτηριστικά των γενετικών αλγορίθμων είναι ότι μπορούν να εφαρμοστούν σε διάφορους τομείς. Οποιοδήποτε και να είναι το πεδίο χρήσης τους όμως αλλά και όποιες αλλαγές έχουν υποστεί προκειμένου να έχουν καλύτερη χρήση τα βασικά βήματα των αλγορίθμων αυτών παραμένουν τα ίδια.

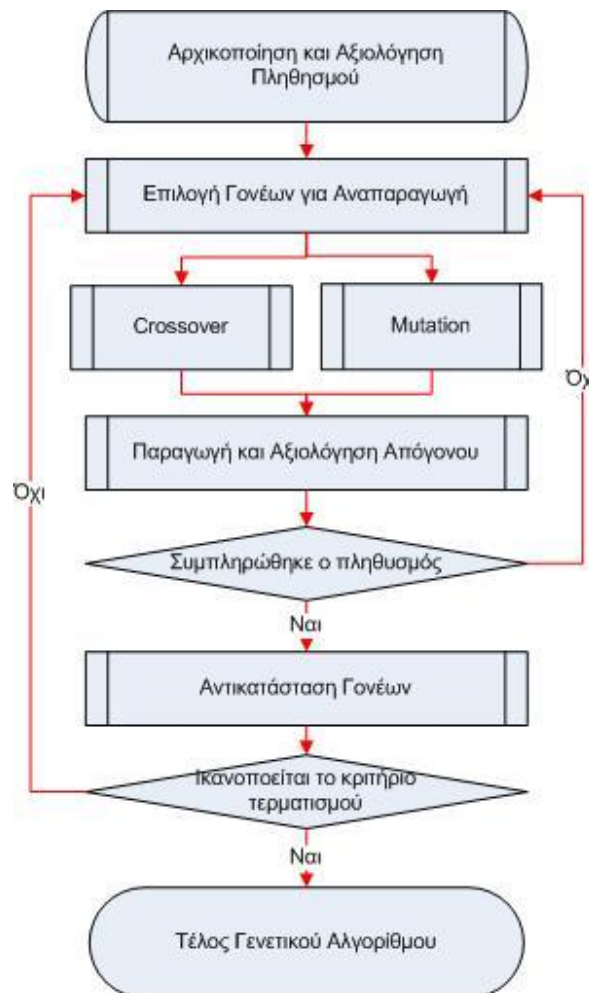
Το πρώτο βήμα είναι να γίνει μια αρχικοποίηση του αλγορίθμου, όλες οι παράμετροι να πάρουν αρχικές τιμές και να δημιουργηθεί ένας αρχικός πληθυσμός πάνω στον οποίο θα βασιστεί στα επόμενα βήματα για να μπορέσει να βρει την καλύτερη δυνατή λύση. Η αρχικοποίηση αυτή είναι τυχαία. Βασικό στοιχείο αποτελούν και ο καθορισμός των γονέων με βάση τους οποίους θα μπορέσει να αρχίσει η επιλογή των απογόνων για αναπαραγωγή. Η επιλογή των γονέων συνήθως είναι αναλογική προς την απόδοσή τους. Υπάρχουν πολλές τεχνικές επιλογής γονέων, με πιο γνωστές αλλά και ευρέως διαδεδομένες αυτές της μεθόδου της ρουλέτας (roulette) και αυτής του τουρνουά (tournament).

Με βάση αυτή την αρχική γενιά θα πρέπει με την βοήθεια γενετικών τελεστών να βρούμε και να παράξουμε τους απογόνους της γενιάς αυτής. Οι πιο διαδεδομένοι τελεστές είναι ο τελεστής της διασταύρωσης (crossover) και ο τελεστής της μετάλλαξης (mutation). Η διασταύρωση συνδυάζει τα στοιχεία των χρωματοσωμάτων δύο γονέων για να δημιουργήσει δύο νέους απογόνους ανταλλάσσοντας κομμάτια από τους γονείς, ενώ η διαδικασία της μετάλλαξης αλλάξει αυθαίρετα ένα ή περισσότερα γονίδια ενός χρωμοσώματος. Να σημειωθεί εδώ ότι τα τελευταία χρόνια έχουν παραχθεί και πολλές παραλλαγές αυτών των δύο τελεστών.

Με τα πιο πάνω, οι γενετικοί αλγόριθμοι δημιουργούν ένα σύνολο - πληθυσμό από πιθανές λύσεις τις οποίες θα πρέπει να αποτιμήσει έτσι ώστε να επιλέξει από το σύνολο αυτό την καλύτερη δυνατή λύση. Αυτή η αποτίμηση γίνεται μέσω μιας συνάρτησης καταλληλότητας (fitness function) η οποία ανάλογα με το πρόβλημα και το χώρο στον οποίο βρισκόμαστε καθορίζεται από τον χρήστη.

Αυτή είναι μια επαναληπτική διαδικασία η οποία τερματίζει μόνο όταν ικανοποιηθεί το κριτήριο τερματισμού που έχει τεθεί. Το πιο συνηθισμένο κριτήριο τερματισμού είναι να έχουμε δημιουργήσει ένα συγκεκριμένο αριθμό γενιών, ενώ χρησιμοποιείται επίσης και το κριτήριο ο αλγόριθμος να φτάσει ένα συγκεκριμένο ποσοστό βελτίωσης του καλύτερου ατόμου ή πληθυσμού.

Όλα τα πιο πάνω βήματα των γενετικών αλγορίθμων φαίνονται και στο πιο κάτω διάγραμμα ροής:



Εικόνα 3.1: Διάγραμμα Ροής Γενετικών Αλγορίθμων [28]

3.4 Πλεονεκτήματα – Μειονεκτήματα Γενετικών Αλγορίθμων

Λόγω του ότι οι αλγόριθμοι αυτοί δεν έχουν μαθηματικό υπόβαθρο αλλά στηρίζονται στην βιολογία και στην έννοια της εξέλιξης - μετάλλαξης, οι γενετικοί αλγόριθμοι έχουν τα εξής πλεονεκτήματα [8], [21]:

- Μπορούν να βρουν τη βέλτιστη λύση σε προβλήματα που δεν είναι δυνατόν να αναλυθούν με μαθηματικό τρόπο.
- Έχουν μεγάλη ελευθερία στην επιλογή της λύσης αφού στηρίζονται κατά πολύ στην έννοια της τυχαίας επιλογής.
- Μπορούν να λύσουν δύσκολα προβλήματα με εύκολο και αξιόπιστο τρόπο: Οι γενετικοί αλγόριθμοι έχουν μεγάλη αποδοτικότητα και αυτό έχει αποδειχτεί όχι μόνο θεωρητικά αλλά και πρακτικά αφού σε προβλήματα με πολλές αλλά και δύσκολες λύσεις οι γενετικοί αλγόριθμοι μπορούν να τα αντιμετωπίσουν καλύτερα από οποιαδήποτε άλλη τεχνική.
- Είναι επαναχρησιμοποιήσιμοι και μπορούν να συνδυαστούν με οποιοδήποτε άλλο σύστημα είτε σαν προ-επεξεργασία ενός προβλήματος είτε σαν ανάλυση αποτελεσμάτων. Μπορείς να χρησιμοποιήσεις ένα γενετικό αλγόριθμο σε υπάρχοντα μοντέλα με προσθετικό τρόπο χωρίς να χρειάζεται να επανασχεδιάσεις ένα νέο γενετικό αλγόριθμο.
- Μπορείς με εύκολο τρόπο να κάνεις αλλαγές αλλά και να κάνεις επεκτάσεις σε ένα γενετικό αλγόριθμο έτσι ώστε να λειτουργεί και να παράγει καλύτερα αποτελέσματα.
- Μπορούν να χειριστούν προβλήματα με πολλές διαστάσεις (παραμέτρους). Οι γενετικοί αλγόριθμοι έχουν το στοιχείο του παραλληλισμού αφού σε κάθε τους βήμα επεξεργάζονται μεγάλες ποσότητες πληροφορίας αφού κάθε άτομο μπορεί να θεωρηθεί σαν ένας αντιπρόσωπος πολλών άλλων. Επομένως μπορούν να κάνουν αποδοτική αναζήτηση σε μεγάλους χώρους σε σχετικά σύντομο χρονικό διάστημα.

- Δεν επηρεάζονται από την φύση ή η σημασία της υπό εξέταση πληροφορίας δηλαδή αν τα δεδομένα μας αφορούν εύρεση συντομότερου μονοπατιού ή εύρεση λύσης που να μειώνει το κόστος. Το μόνο τους μέλημα είναι η συνάρτηση αξιολόγησης.
- Μπορούν να συνδυαστούν με άλλες μεθόδους δημιουργώντας με αυτό το τρόπο υβριδικά συστήματα.
- Κάνουν χρήση της πληροφορίας που έχουν ήδη παράξει προκειμένου να δώσουν καλύτερη λύση αλλά ταυτόχρονα γίνεται και εξερεύνηση του χώρου αναζήτησης, δύο χαρακτηριστικών τα οποία είναι εκ γενετής ανταγωνιστικά και δύσκολα συνυπάρχουν.

Εκτός όμως από τα πολλά πλεονεκτήματα που παρουσιάζουν οι γενετικοί αλγόριθμοι εντούτοις όπως και όλες οι εφαρμογές παρουσιάζουν και κάποια μειονεκτήματα που για μερικούς μπορεί να είναι αποτρεπτικά. Κάποια από αυτά τα μειονεκτήματα παρουσιάζονται πιο κάτω [21]:

- Μεγάλος χρόνος επεξεργασίας. Τέτοιες εφαρμογές χρειάζονται αρκετό χρόνο μέχρι να περάσουν από πολλές γενιές και να χρησιμοποιήσουν την θεωρία της εξέλιξης για να παράξουν αποτελέσματα. Η εξέλιξη λειτουργεί με πολύ αργούς ρυθμούς και χρειάζονται χιλιάδες γενιές για να αλλάξουν τα χαρακτηριστικά των ειδών και να διαφοροποιηθεί η συμπεριφορά τους.
- Ο αφαιρετικός τρόπος μίμησης τέτοιων συστημάτων προς την βιολογία ίσως κάποιες φορές να είναι ανασταλτικός παράγοντας εφόσον το να μην είναι κάποιος οικείος με τέτοιες έννοιες όπως είναι οι έννοιες της εξέλιξης και της φυσικής επιλογής, μπορεί να προκαλέσει φόβο.

3.5 Εφαρμογές Γενετικών Αλγορίθμων

Με βάση όλα τα πιο πάνω χαρακτηριστικά που έχουμε αναφέρει για τους γενετικούς αλγόριθμους είναι λογικό συμπέρασμα να γίνεται με ευρεία χρήση και εφαρμογή των

αλγορίθμων αυτών. Μερικές από τις εφαρμογές των γενετικών αλγορίθμων είναι και οι ακόλουθες [9]:

- Εξαγωγή συμπερασμάτων για την διαδικασία υπολογισμού του κόστους μιας καινούργιας εφαρμογής.
- Πρόβλημα πλανόδιου πωλητή, καταμερισμός εργασιών κ.τ.λ.
- Επεξεργασία εικόνων, σήματος και αναγνώριση προτύπων.
- Προβλήματα κατανομής πόρων με τέτοιο τρόπο έτσι ώστε να έχουμε μέγιστο όφελος και ελάχιστο κόστος.
- Εύρεση λύσεων σε παιχνίδια και επίλυση λαβυρίνθων.
- Εκπαίδευση Νευρωνικών δικτύων.
- Βελτιστοποίηση μαθηματικών μοντέλων που περιγράφουν ένα οποιοδήποτε πρόβλημα φυσικό, μαθηματικής λογικής, βιολογίας, στατιστικής, διοίκησης επιχειρήσεων κ.τ.λ.

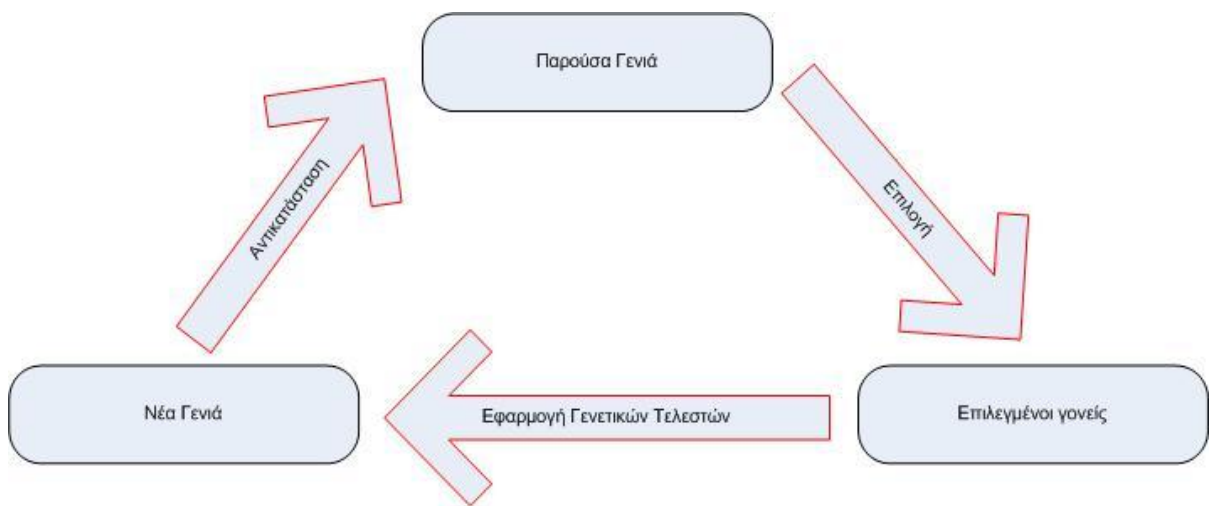
3.6 Γενετικός Προγραμματισμός

Ο γενετικός προγραμματισμός ίσως να είναι η πιο δημοφιλής τεχνική της ομάδας των εξελικτικών αλγορίθμων και αποτελεί τεχνική για την υπολογιστική προσομοίωση της εξέλιξης. Ουσιαστικά αποτελεί προέκταση των γενετικών αλγορίθμων αφού αφαιρεί τον περιορισμό ότι κάθε άτομο πρέπει να έχει ένα προκαθορισμένο μέγεθος. Η πιο συνηθισμένη αλλά και απλή μέθοδος αναπαράστασης σε ένα γενετικό προγραμματισμό είναι ένα δυαδικό δέντρο το οποίο αποτελείται από τελεστές και τελεστέους. Με αυτό το τρόπο κάθε λύση είναι μια αλγεβρική έκφραση η οποία σε τελική ανάλυση μπορεί να αποτιμηθεί.

Σε ένα γενετικό πρόγραμμα το πρώτο πράγμα που πρέπει να γίνει είναι να καθορίσουμε πιο θα είναι το αλφάβητο, ποιοι θα είναι οι τελεστές και ποιοι οι τελεστέοι. Ένα δεύτερο βήμα είναι η δημιουργία του αρχικού πληθυσμού. Συνήθως αυτό γίνεται με τυχαία

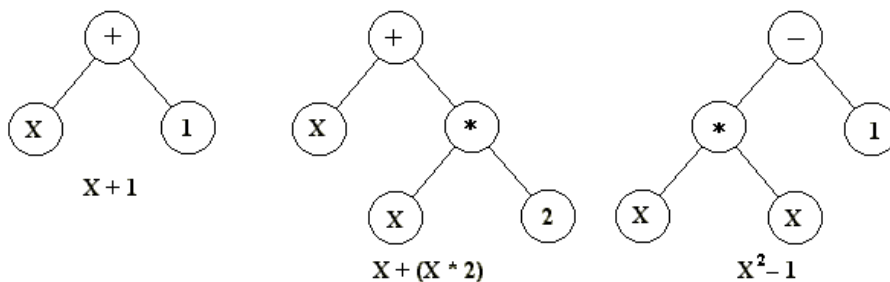
επιλογή από το σύνολο των τελεστών και των τελεστέων. Στη συνέχεια γίνεται επιλογή γονέων και με χρήση γενετικών τελεστών θα δημιουργηθεί μια νέα γενιά. Πολύ σημαντική είναι και η έννοια της τυχαίας επιλογής. Σε όλα τα βήματα του γενετικού προγραμματισμού όλες οι επιλογές που γίνονται είναι με τυχαίο τρόπο, είτε είναι τελεστής, είτε είναι τελεστέος είτε είναι γενετικός τελεστής.

Ο βασικός κύκλος εξέλιξης σε ένα γενετικό πρόγραμμα παρουσιάζεται πιο κάτω στο επόμενο διάγραμμα.



Εικόνα 3.2: Κύκλος Εξέλιξης Γενετικού Προγραμματισμού [28]

Όπως έχουμε ήδη αναφέρει, εκτός από την έννοια της τυχαίας επιλογής, βασικό στοιχείο του προγραμματισμού αυτού είναι η δημιουργία δέντρων για την αναπαράσταση της λύσης. Ένα τέτοιο δέντρο είναι και το πιο κάτω:



Εικόνα 3.3: Μορφή Δέντρου Γενετικού Προγραμματισμού

3.6.1 Συνάρτηση Καταλληλότητας

Η συνάρτηση καταλληλότητας είναι από τα βασικότερα στοιχεία του γενετικού προγραμματισμού που ουσιαστικά δίνει το μέτρο ικανότητας της λύσης. Κάθε μια λύση που παράγεται από το σύστημα δοκιμάζεται για το πόσο κοντά φέρνει την αντίδραση του συστήματος στην επιθυμητή τιμή. Συνήθως η τιμή καταλληλότητας αποτελεί τον καθοριστικό παράγοντα επιβίωσης του κάθε ατόμου. Ανάλογα με τον τρόπο που θα ορίσει κανείς αυτή τη συνάρτηση εξαρτάται αν ο γενετικός προγραμματισμός θα επιδιώξει να έχει ελάχιστη ή μέγιστη τιμή για την επιλογή του καλύτερου ατόμου. Η συνάρτηση καταλληλότητας θέλουμε να είναι όσο το δυνατό πιο απλή και εύκολα υπολογίσιμη.

3.6.2 Παράγοντας bloating

Ένα πολύ συχνό φαινόμενο που παρατηρείται στο γενετικό προγραμματισμό είναι το φαινόμενο του “bloating”. Αυτό συμβαίνει όταν υπάρχει μια τάση το μέγεθος των δέντρων που παράγονται να αυξάνεται συνεχώς χωρίς ουσιαστική βελτίωση στην τιμή της συνάρτησης καταλληλότητας και ως αποτέλεσμα να μην μπορούμε να έχουμε την καλύτερη αλλά και συνάμα την πιο μικρή σε μέγεθος δυνατή λύση. Αυτό το φαινόμενο προκύπτει από την αναζήτηση που κάνουν οι γενετικοί τελεστές στο να βρουν την καλύτερη δυνατή λύση. Εκτός από το γεγονός ότι καταναλώνεται πολύτιμος χρόνος σε μια ήδη πολύ βαριά σε επεξεργασία διαδικασία, αυτά τα περιττά κομμάτια κώδικα (introns) αυξάνονται ραγδαία.

Παρόλο που το φαινόμενο αυτό είναι γνωστό και μπορεί να εντοπιστεί, εντούτοις δεν υπάρχει ακόμα εξήγηση γιατί να συμβαίνει αλλά ούτε επίσης έχει προταθεί μια πρακτική λύση για να αποφεύγονται τέτοια φαινόμενα. Υπάρχουν όμως αρκετές θεωρίες στο τι μπορεί να προκαλεί αυτές τις ανεπιθύμητες καταστάσεις [26]:

- Η θεωρία του περιττού κώδικα (introns theory). Σε αυτή την θεωρία πιστεύεται ότι τα κομμάτια αυτά του περιττού κώδικα που παράγονται, δημιουργούν bloating σαν ένας προστατευτικός μηχανισμός έτσι ώστε να αποφευχθούν τυχόν καταστρεπτικές παρενέργειες των τελεστών όταν βρεθεί μια λύση. Να

σημειώσουμε εδώ ότι αυτά τα κομμάτια κώδικα δεν επηρεάζουν την συνάρτηση καταλληλότητας (fitness).

- Η θεωρία του προκαλούμενου από τη συνάρτηση καταλληλότητας (fitness causes bloat theory). Αυτή η θεωρία βασίζεται στην υπόθεση ότι υπάρχει μεγαλύτερη πιθανότητα να βρεθεί μεγαλύτερο πρόγραμμα με την ίδια συμπεριφορά παρά ένα μικρότερο σε μέγεθος πρόγραμμα. Ως αποτέλεσμα όταν μια καλή λύση βρεθεί τα προγράμματα έχουν την τάση να μεγαλώνουν λόγω της πίεσης της συνάρτησης καταλληλότητας.
- Η θεωρία της αφαίρεσης της προκατάληψης (removal bias theory). Εδώ ουσιαστικά επικρατεί η προκατάληψη ότι αφαιρώντας μεγάλα κομμάτια κώδικα είναι πιο επικίνδυνο από το να αφαιρέσεις μικρά κομμάτια κώδικα. Έτσι με αυτό τον τρόπο υπάρχει μια προκατάληψη υπέρ των μεγάλων προγραμμάτων.

Κεφάλαιο 4

Προτεινόμενο Σύστημα και Πειραματική Διαδικασία

- 4.1 Εισαγωγή
 - 4.2 Εργαλείο GPLAB
 - 4.3 Περιγραφή Προτεινόμενου Συστήματος
 - 4.3.1 Επιλογή Αρχείου Δεδομένων
 - 4.3.2 Επιλογή Παραμέτρων
 - 4.3.3 Εκτέλεση Γενετικού Προγραμματισμού
 - 4.3.4 Δημιουργία Αρχείου Εξόδου
 - 4.3.5 Παραγόμενες Γραφικές Παραστάσεις
-

4.1 Εισαγωγή

Για τις ανάγκες της παρούσας μελέτης έχουμε κάνει χρήση μιας εφαρμογής η οποία μας επιτρέπει να κάνουμε αλλαγές αλλά και να προσθέσουμε συναρτήσεις εύκολα έτσι ώστε να προσαρμόσουμε την εκτέλεση με βάση τα δικά μας δεδομένα και γενικά τις δικές μας ανάγκες. Αυτή η εφαρμογή είναι το GPLAB το οποίο είναι γραμμένο στη γλώσσα προγραμματισμού MATLAB[®] της εταιρείας Math Works που είναι μια υψηλής απόδοσης γλώσσα (4^{ης} γενιάς) που χρησιμοποιείται για Technical Computing και η οποία μπορεί να εκφράζει λύσεις στα προβλήματα με μαθηματικό τρόπο. Είναι γλώσσα απλή και εύκολη στη χρήση αλλά παράλληλα γρήγορη και δυνατή όσον αφορά μαθηματικούς υπολογισμούς. Ένα άλλο δυνατό πλεονέκτημα της γλώσσας αυτής είναι το γεγονός ότι παρέχει διεξοδική αλλά και απλή αναπαράσταση δεδομένων.

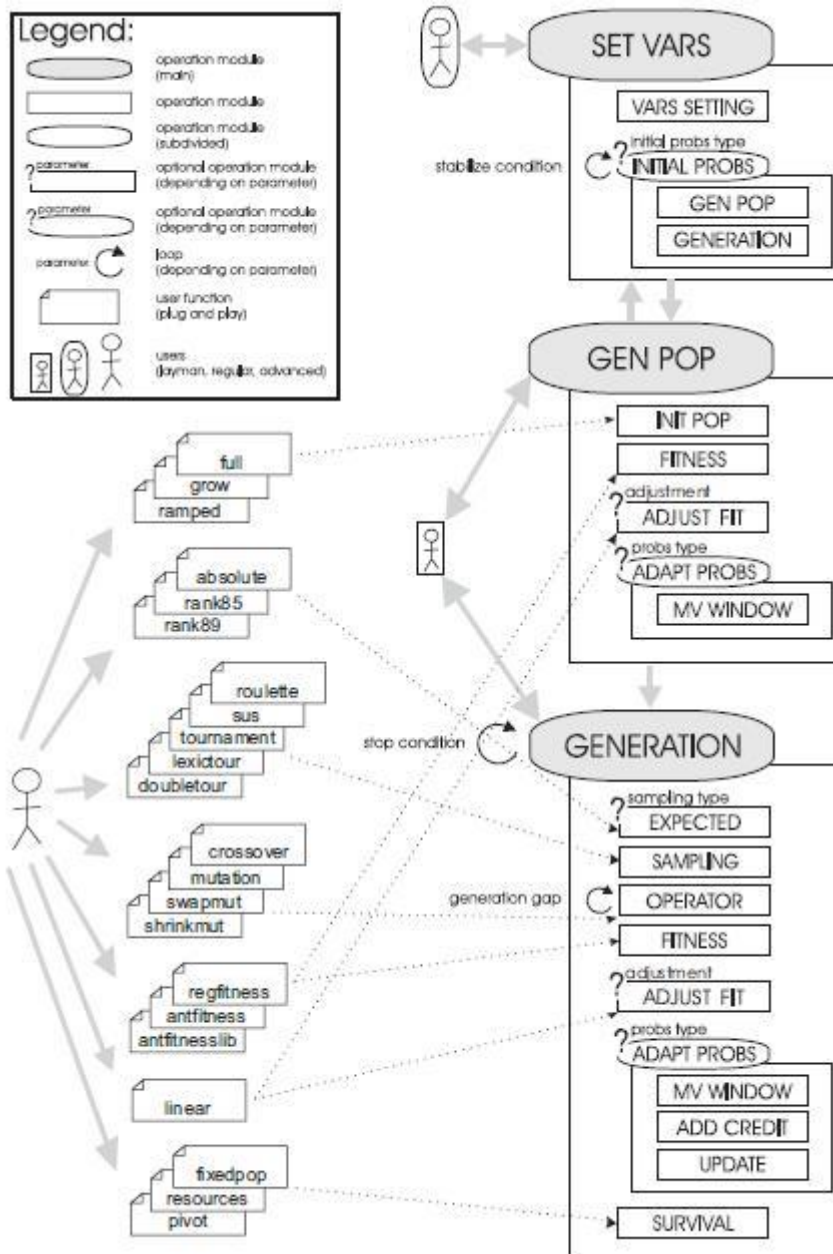
4.2 Εργαλείο GPLAB

Το GPLAB [24] όπως λέει και το όνομα του, είναι εργαλείο γενετικού προγραμματισμού. Είναι γραμμένο σε γλώσσα προγραμματισμού MATLAB, γενικευμένο, προσαρμοστικό αλλά και εύκολα επεκτεινόμενο εργαλείο. Ο σχεδιασμός και η υλοποίηση του έχουν γίνει

με τέτοιο τρόπο ώστε να μπορεί να χρησιμοποιηθεί όχι μόνο από άτομα που έχουν βαθιά γνώση από γενετικό προγραμματισμό ή προγραμματισμό αλλά και από άτομα που δεν έχουν την αντίστοιχη εξοικείωση. Η αρχιτεκτονική του εργαλείου αυτού είναι αρθρωτή και αποτελείται από πολλές υπό μονάδες και έχει παραμετρική δομή. Το μεγαλύτερο πλεονέκτημα του όμως είναι το γεγονός ότι ο κάθε χρήστης μπορεί να δημιουργήσει δικές του λειτουργίες και να τις προσθέσει στο εργαλείο αυτό.

Το εργαλείο αποτελείται από τρεις βασικές μονάδες: την μονάδα καθορισμού των μεταβλητών, την μονάδα δημιουργίας του πληθυσμού και την μονάδα δημιουργίας γενεών. Και οι τρεις αυτές μονάδες έχουν άμεση σχέση με μεταβλητές που πρέπει να επιλεχθούν από το χρήστη. Η μονάδα καθορισμού των μεταβλητών έχει στόχο την αρχικοποίηση όλων των απαραίτητων παραμέτρων που θα καθορίσει ο χρήστης προκειμένου να μπορέσει το εργαλείο να εκτελέσει τις λειτουργίες του. Για παραμέτρους όπου ο χρήστης δεν δώσει κάποια τιμή το εργαλείο παρέχει από το σχεδιασμό του προκαθορισμένες τιμές τις οποίες και δίνει σε αυτές τις περιπτώσεις. Η δεύτερη μονάδα δημιουργεί τον αρχικό πληθυσμό και υπολογίζει την ικανότητα του πληθυσμού αυτού (fitness). Στο εργαλείο αυτό τα άτομα του πληθυσμού είναι αναπαραστάσεις με μορφή δένδρων. Τέλος στην μονάδα δημιουργίας γενεών, έχουμε τη δειγματοληψία (sampling) για να δημιουργήσουμε την λίστα με τους διαθέσιμους γονείς ενώ επίσης έχουμε και την εφαρμογή των γενετικών τελεστών προκειμένου να δημιουργήσουμε τα νέα άτομα. Με την δημιουργία των νέων ατόμων έχουμε υπολογισμό της ικανότητας του πληθυσμού που έχουμε δημιουργήσει και στη συνέχεια έχουμε το στάδιο της επιβίωσης των καλύτερων ατόμων (survival). Το τελευταίο αυτό στάδιο επαναλαμβάνεται μέχρι να συναντήσουμε κάποιο από τα κριτήρια τερματισμού ή μέχρι να φτάσουμε στον μέγιστο αριθμό γενιών που έχουμε θέσει.

Πιο κάτω μπορούμε να δούμε την διαγραμματική απεικόνιση που έχουμε περιγράψει πιο πάνω.



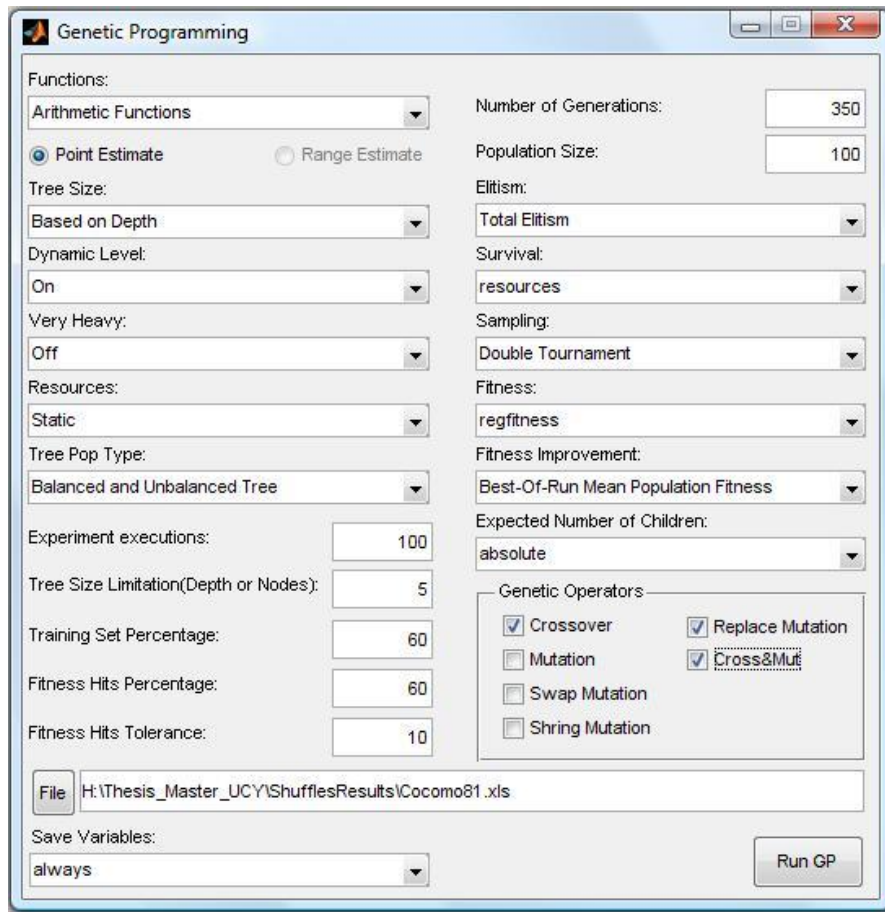
Εικόνα 4.1: Λειτουργική δομή εργαλείου GPLAB [24]

Το πιο πάνω διάγραμμα δείχνει την σχεδιασμένη λειτουργία του εργαλείου καθώς και τις βασικές διαθέσιμες λειτουργίες που μπορεί να επιλέξει ο χρήστης σε κάθε βήμα. Για την αρχικοποίηση του πληθυσμού παρέχονται οι επιλογές για πλήρως ισοροπημένα δέντρα (full), για πλήρως ανισόροπα δέντρα (grow) ή για μικτά δέντρα (ramped). Για την παραγωγή των παιδιών έχουμε και εδώ τρεις επιλογές. Τα παιδιά για κάθε άτομο να είναι αναλογικά της ικανότητας του ατόμου (absolute), τα παιδιά βασίζονται στην κατάταξη του ατόμου (rank85) ή τα παιδιά βασίζονται όχι μόνο στην κατάταξη του

ατόμου αλλά και στην κατάσταση του αλγορίθμου την χρονική στιγμή της επιλογής (rank89). Στην φάση της δειγματοληψίας παρέχονται πέντε διαφορετικές συναρτήσεις. Στη ρουλέτα (roulette) έχουμε τυχαίους δείχτες να γυρίζουν και κάθε άτομο της γενιάς να έχει ένα τυχαίο κομμάτι αυτής της ρουλέτας που αντιπροσωπεύει τους απογόνους. Μια παραλλαγή είναι η sus στην οποία οι δείχτες της ρουλέτας είναι ίσα κατανεμημένοι. Στο τουρνουά (tournament), επιλέγεται τυχαία ένας αριθμός από άτομα και μέσα από αυτό τον αριθμό επιλέγονται τα καλύτερα άτομα. Η μέθοδος της λεξικογραφίας (lexicographic parsimony pressure) δουλεύει όπως και το τουρνουά αλλά με την διαφορά ότι όταν δύο άτομα έχουν την ίδια συνάρτηση καταλληλότητας θα επιλεγεί το πιο μικρό σε μέγεθος. Η τελευταία συνάρτηση για δειγματοληψία είναι το διπλό τουρνουά (double tour) το οποίο συνδυάζει την μέθοδο τουρνουά και μετά εφαρμόζει τη μέθοδο της λεξικογραφίας. Ο υπολογισμός της συνάρτησης καταλληλότητας (fitness) μπορεί να γίνει με τη μέθοδο που ονομάζεται regfitness η οποία ουσιαστικά υπολογίζει για κάθε άτομο τη διαφορά της πραγματικής τιμής από την υπολογιζόμενη. Τέλος για να επιλεγούν τα καλύτερα άτομα (survival) έχουμε τρεις συναρτήσεις. Η συνάρτηση fixedprosize ουσιαστικά δηλώνει ότι θα επιλεγούν τόσα άτομα όσα και τα άτομα του πληθυσμού μας. Με αυτό τον τρόπο θα έχουμε συνεχώς σταθερό αριθμό ατόμων στον πληθυσμό. Η συνάρτηση resources επιτρέπει να έχουμε αριθμό ατόμων όχι απαραίτητα τον ίδιο με τον αρχικό πληθυσμό. Το GPLAB έχει δημιουργήσει ακόμα μια συνάρτηση, την rinotfixe, η οποία κάνει το ίδιο με την resources αλλά έχει σκοπό να εξοικονομήσει πόρους και υπολογιστική προσπάθεια του συστήματος.

4.3 Περιγραφή Προτεινόμενου Συστήματος

Το προτεινόμενο σύστημα βασίζεται στις διάφορες συναρτήσεις που παρέχει το πιο πάνω εργαλείο για γενετικό προγραμματισμό. Έχει δημιουργηθεί με τέτοιο τρόπο έτσι ώστε να είναι εύχρηστο και λειτουργικό και να παρέχει στο χρήστη όσο το δυνατό περισσότερες επιλογές για τις εκτελέσεις του. Όπως και το GPLAB έτσι και αυτό το σύστημα έχει σχεδιαστεί και υλοποιηθεί στην γλώσσα προγραμματισμού MATLAB. Ο χρήστης μπορεί μέσω μιας φιλικής διεπαφής (interface) να δημιουργήσει τις επιθυμητές εκτελέσεις μέσω των προσφερόμενων επιλογών παραμέτρων που έχουν δοθεί στο προτεινόμενο αυτό σύστημα. Η διεπαφή αυτή φαίνεται πιο κάτω στην εικόνα 4.2.



Εικόνα 4.2: Προτεινόμενο Σύστημα

Όπως φαίνεται και στην οθόνη του προτεινόμενου συστήματος ο χρήστης καλείται να επιλέξει μια σειρά από παραμέτρους προκειμένου να διοργανώσει τα πειράματα που επιθυμεί. Το σύστημα αποτελείται από τέσσερα βασικά μέρη:

1. Επιλογή αρχείου δεδομένων
2. Επιλογή παραμέτρων
3. Εκτέλεση γενετικού προγραμματισμού
4. Δημιουργία αρχείου εξόδου
5. Παραγόμενες γραφικές παραστάσεις

Ας δούμε λίγο πιο αναλυτικά το κάθε ένα από αυτά.

4.3.1 Επιλογή Αρχείου Δεδομένων

Το προτεινόμενο σύστημα δέχεται ως είσοδο του ένα αρχείο δεδομένων σε μορφή Excel. Στο αρχείο αυτό θα πρέπει η πρώτη γραμμή να περιέχει τα ονόματα των μεταβλητών όπως θέλουμε να φαίνονται στα δένδρα που θα παραχθούν αλλά και στις παραγόμενες εξισώσεις. Το αρχείο δεν έχει περιορισμό στον αριθμό των στηλών που πρέπει να έχει, όμως υπάρχει η προϋπόθεση η τελευταία στήλη να είναι το χαρακτηριστικό που αντιπροσωπεύει την πραγματική τιμή της προσπάθειας. Για τις αριθμητικές αλλά και τις λογικές εκτελέσεις θα πρέπει το αρχείο να αποτελείται από αριθμητικά δεδομένα, ενώ για τις κατηγορικές εκτελέσεις θα πρέπει να έχουμε κατηγορικά δεδομένα στο αρχείο αυτό. Όλα τα χαρακτηριστικά που υπάρχουν στα αρχεία δεδομένων που θα χρησιμοποιήσουμε ονομάζονται και παράγοντες κόστους.

Στην παρούσα μελέτη θα χρησιμοποιήσουμε τρία αρχεία δεδομένων, το αρχείο Cocomo [6], το αρχείο Desharnais [15] και το αρχείο ISBSG [12]. Το αρχείο του COCOMO (εικόνα 4.3) αποτελείται από 16 χαρακτηριστικά τα οποία είναι όλα αριθμητικά δεδομένα και 63 έργα ανάπτυξης λογισμικού, τα οποία λήφθηκαν από έργα ανάπτυξης ενός Οίκου Ανάπτυξης Λογισμικών.

1.	LOC	Αριθμός Γραμμών Κώδικα
2.	RELY	Απαιτούμενη αξιοπιστία λογισμικού
3.	DATA	Μέγεθος βάσης δεδομένων
4.	CPLX	Πολυπλοκότητα προϊόντος
5.	TIME	Περιορισμός στο χρόνο εκτέλεσης
6.	STOR	Περιορισμός στην κύρια μνήμη
7.	VIRT	Αλλαγές στο σύστημα HW/SW
8.	TURN	Χρόνος απόκρισης υπολογιστή
9.	ACAP	Ικανότητα αναλυτών
10.	AEXP	Εμπειρία αναυτών σε εφαρμογές
11.	PCAP	Ικανότητα προγραμματιστών
12.	VEXP	Εμπειρία με το σύστημα HW/SW
13.	LEXP	Εμπειρία με τη γλώσσα προγραμματισμού
14.	MODP	Χρήση μοντέρνων πρακτικών προγραμματισμού
15.	TOOL	Χρήση εργαλείων προγραμματισμού
16.	SCED	Πίεση από χρονοδιάγραμμα ανάπτυξης

Εικόνα 4.3: Χαρακτηριστικά αρχείου εισόδου Cocomo [6]

Με το αρχείο αυτό λόγω του γεγονότος ότι έχουμε μόνο αριθμητικά δεδομένα, μπορούμε να τρέξουμε μόνο αριθμητικές και λογικές εκτελέσεις.

Το αρχείο Desharnais (εικόνα 4.4) αποτελείται από 9 χαρακτηριστικά εκ των οποίων το ένα είναι κατηγορικό ενώ τα υπόλοιπα είναι αριθμητικά και περιέχει παρατηρήσεις για 77 συστήματα ανάπτυξης, τα οποία έχουν παρθεί από ένα Καναδικό Οίκο Ανάπτυξης Λογισμικού στα τέλη της δεκαετίας του 1980.

1.	TEXP	Εμπειρία Ομάδας
2.	MEXP	Εμπειρία Χειριστή
3.	LENGTH	Διάρκεια σε Μήνες
4.	TRANS	Συναλλαγές
5.	ENTS	Οντότητες
6.	PADJ	Προσαρμοσμένα Σημεία
7.	ENV	Πλαίσιο
8.	PNADJ	Μη Προσαρμοσμένα Σημεία
9.	LANG	Γλώσσα

Εικόνα 4.4: Χαρακτηριστικά αρχείου εισόδου Desharnais [15]

Για τις αριθμητικές και τις λογικές εκτελέσεις με αυτό το αρχείο θα αφαιρέσουμε την τελευταία στήλη που αντιστοιχεί στη γλώσσα προγραμματισμού αφού είναι μια κατηγορική στήλη.

Το τρίτο αρχείο που έχουμε χρησιμοποιήσει για τις εκτελέσεις μας είναι το αρχείο ISBSG (εικόνα 4.5). Το αρχείο αυτό είναι αρκετά μεγάλο σε σχέση με τα προηγούμενα δύο αρχεία αφού αποτελείται από 467 έργα λογισμικού και 50 χαρακτηριστικά ένα εκ των οποίων είναι και η πραγματική τιμή της προσπάθειας. Από τα χαρακτηριστικά αυτά έχουμε αφαιρέσει τα 4 που είναι αριθμητικά για να μπορέσουμε να έχουμε ένα σύνολο από κατηγορικά δεδομένα και έτσι να μπορέσουμε να το χρησιμοποιήσουμε σε κατηγορικές εκτελέσεις.

1 CP	Count Approach	26 EMODEL	Event Modelling
2 AFP	Adjusted Function Points	27 BAM	Business Area Modelling
3 PET	Project Elapsed Time	28 OOA	Object Oriented Analysis
4 IYEAR	Implementation Year	29 TBOX	Timeboxing
5 DEVT	Development Type	30 STDS	Standards
6 AEAU	Aerospace / Automotive	31 WATERF	Waterfall
7 BANK	Banking	32 ODT	OTHER DEVELOPMENT TECHNIQUE
8 COMM	Communications	33 FST	Functional Sizing Technique
9 EGW	Electricity, Gas, Water	34 DEVP	Development Platform
10 FPBS	Financial, Property & Business Services	35 LTYPE	Language Type
11 GOV	Government	36 PPL	Primary Programming Language
12 INS	Insurance	37 DB2	DB2
13 MAN	Manufacturing	38 IMS	IMS
14 PADM	Public Administration	39 ORACLE	ORACLE
15 TRST	Transport & Storage	40 ADABAS	ADABAS
16 WRT	Wholesale & Retail Trade	41 VSAM	VSAM
17 OOT	OTHER ORGANIZATION TYPE	42 SYBASE	SYBASE
18 DMODEL	Data Modelling	43 ACCESS	ACCESS
19 PROTO	Prototyping	44 SQLSRV	SQL SERVER
20 PMODEL	Process Modelling	45 O1DBS	OTHER 1ST DBASE SYSTEM
21 JAD	JAD (Joint Application Development)	46 RECM	Recording Method
22 MTEAM	Multifunctional Teams	47 RESL	Resource Level
23 OOD	Object Oriented Design	48 MAXTS	Max Team Size
24 RAD	RAD (Rapid Application Development)	49 AVGTS	Average Team Size
25 REGT	Regression Testing	50 Effort	Full Cycle Work Effort

Εικόνα 4.5: Χαρακτηριστικά αρχείου εισόδου ISBSG [12]

4.3.2 Επιλογή παραμέτρων

Υπάρχουν διάφοροι παράμετροι με τους οποίους μπορούμε να κάνουμε διάφορες εκτελέσεις και να συγκρίνουμε αποτελέσματα. Αυτός είναι και ο λόγος που επιλέξαμε όλες αυτές οι παράμετροι να είναι καθορισμένες από το χρήστη σε κάθε εκτέλεση.

Ας δούμε αναλυτικά μια προς μια τις παραμέτρους αυτές οι οποίες φαίνονται και στην εικόνα 4.2 πιο πάνω. Βλέποντας την εικόνα 4.2, μπορούμε να διακρίνουμε ότι το δεξιό μέρος της αφορά παραμέτρους που έχουν να κάνουν απευθείας με τους γενετικούς αλγορίθμους, ενώ το αριστερό κομμάτι είναι μεταβλητές που υποστηρίζουν την όλη διαδικασία. Θα αποκαλούμε την πρώτη ομάδα ως παράμετροι γενετικού αλγορίθμου ενώ η δεύτερη ομάδα ως υποστηρικτικοί παράμετροι γενετικού αλγορίθμου.

4.3.2.1 Παράμετροι Γενετικού Αλγορίθμου

Αριθμός Γενιών (Number of Generations): Δηλώνει τον αριθμό των γενιών που θα τρέξει ο αλγόριθμος προκειμένου να βγάλει το καλύτερο άτομο. Για τα πειράματά μας θα χρησιμοποιήσουμε 350 γενιές και αν δούμε ότι είναι αναγκαίο με βάση τα παραγόμενα λάθη θα αυξήσουμε τον αριθμό αυτό στο 500.

Μέγεθος πληθυσμού (Population Size): Δηλώνει το μέγεθος του πληθυσμού που πρέπει να έχει κάθε γενιά πριν να αξιολογηθεί και να επιλεγεί το καλύτερο άτομο. Η σωστή επιλογή για το μέγεθος του πληθυσμού είναι πολύ σημαντική αφού μπορεί να καθορίσει κατά πόσο η εκτέλεση μας θα είναι επιτυχής ή όχι. Σαν μέγεθος πληθυσμού στην πειραματική μας διαδικασία θα έχουμε 100.

Ελιτισμός (Elitism): Είναι ο τρόπος με τον οποίο θα δοθεί προτεραιότητα σε κάθε άτομο της γενιάς, είναι ο τρόπος επιλογής των καλύτερων ατόμων με έμφαση στο να επιλεγούν τα καλύτερα άτομα. Είναι μια σημαντική παράμετρος αφού επιτρέπει την λύση μας να γίνεται καλύτερη με την πάροδο του χρόνου. Υπάρχουν διαθέσιμα 4 ήδη ελιτισμού:

- Αντικατάσταση (Replace): Οι απόγονοι αντικαθιστούν τους γονείς γι'αυτό και παίρνουν μεγαλύτερη προτεραιότητα ακόμα και αν είναι χειρότεροι από τους γονείς.
- Επικράτηση Καλύτερου (Keep Best): Το καλύτερο άτομο και από τους γονείς αλλά και από τους απογόνους θα παραμείνει στον πληθυσμό.
- Μισός Ελιτισμός (Half Elitism): Τα καλύτερα μισά άτομα από τους γονείς και τους απογόνους θα παραμείνουν στον νέο πληθυσμό.
- Ολικός Ελιτισμός (Total Elitism): Όλα τα άτομα είτε είναι γονείς είτε απόγονοι παίρνουν προτεραιότητα σύμφωνα με το fitness μόνο.

Στην πειραματική μας διαδικασία θα χρησιμοποιήσουμε το 4^ο είδος ελιτισμού.

Επιβίωση (Survival): Η επιβίωση γίνεται μετά από τον ελιτισμό και έχει να κάνει με το πόσα άτομα θα παραμείνουν στον νέο πληθυσμό που έχει δημιουργηθεί:

- Σταθερό Μέγεθος (fixedpopsize): Ο αριθμός των ατόμων σε ένα πληθυσμό παραμένει ο ίδιος κατά τη διάρκεια του γενετικού αλγορίθμου.
- Βάση Πόρων (Resources) & Pivotfixe: Ο αριθμός των ατόμων σε ένα πληθυσμό μπορεί να αλλάξει ανάλογα με τις ανάγκες. Η τεχνική με βάση τους πόρους εμπειρικά χρησιμοποιείται για να αποφεύγονται φαινόμενα bloating.

Για επιβίωση τα πειράματά μας θα έχουν την τιμή resources.

Δειγματοληψία (Sampling): Είναι η μέθοδος με την οποία θα γίνει η επιλογή των γονέων μιας γενιάς και σε αυτή την περίπτωση έχουμε 4 ήδη δειγματοληψίας:

- Ρουλέτα (Roulette): Η επιλογή των γονέων γίνεται με μια ρουλέτα με τυχαίους δείχτες που γυρίζει και κάθε άτομο έχει το δικό του κομμάτι σε αυτή την ρουλέτα.
- SUS: Η τεχνική αυτή είναι η ίδια με αυτή της ρουλέτας με την διαφορά ότι εδώ οι δείχτες είναι ίσα κατανεμημένοι.
- Τουρνουά (Tournament): Η επιλογή των γονέων γίνεται με τυχαία επιλογή των καλύτερων ατόμων του πληθυσμού.
- Λεξικογραφικά (Lexicography): Αυτή η μέθοδος είναι ίδια με τη μέθοδο τουρνουά αλλά με την διαφορά ότι σε άτομα με ίδιο fitness θα πάρουμε το άτομο που είναι μικρότερο σε μέγεθος. Είναι τεχνική που μπορεί να μας βοηθήσει με το πρόβλημα του bloating.
- Διπλό Τουρνουά (Double Tournament): Το διπλό τουρνουά είναι συνδυασμός της μεθόδου τουρνουά αλλά και της λεξικογραφικής μεθόδου.

Λαμβάνοντας υπόψη όλες τις πιο πάνω μεθόδους εμείς στην πειραματική μας διαδικασία θα χρησιμοποιήσουμε το διπλό τουρνουά.

Συνάρτηση Καταλληλότητας (Fitness): Καθορίζουμε τον τρόπο που θα υπολογίζεται η καταλληλότητα ενός ατόμου μέσα στον πληθυσμό. Εδώ έχουμε

ουσιαστικά την μέθοδο regfitness η οποία υπολογίζει για κάθε άτομο την απόλυτη διαφορά της πραγματικής τιμής του effort με την υπολογίσιμη τιμή του effort.

$$fitness = \sum_{i=1}^n |x_{act}(i) - x_{pred}(i)|$$

Εκτός από αυτή την συνάρτηση καταλληλότητας, έχουμε δημιουργήσει και μια δική μας συνάρτηση η οποία βασίζεται στην τιμή του συντελεστή του μέσου σχετικού σφάλματος.

$$fitness = MRE(n) = \frac{\frac{1}{n} \sum_{i=1}^n |x_{act}(i) - x_{pred}(i)|}{x_{act}(i)}$$

Αυτή η συνάρτηση εφαρμόζεται στις αριθμητικές εκτελέσεις ενώ στις λογικές εκτελέσεις η συνάρτηση καταλληλότητας που εφαρμόσαμε είναι διαφορετική. Σε αυτές τις περιπτώσεις η ικανότητα μετρείται συναρτήσει του αριθμού των γραμμών που ικανοποιούν την εξίσωση, των αριθμών των επιπέδων του κανόνα καθώς και ενός διαστήματος στη τιμή του effort. Με αυτό τον τρόπο χειριζόμαστε το θέμα του bloating, με το να επιλέγονται κανόνες στους οποίους η τυπική απόκλιση της τιμής της προβλεπόμενης προσπάθειας καθώς και το μέγεθος του κανόνα (σε αριθμό κόμβων του παραγόμενου δένδρου) θα είναι ελάχιστο, ενώ παράλληλα ικανοποιούν και μεγάλο αριθμό έργων από το σύνολο των δεδομένων μας. Πιο συγκεκριμένα η συνάρτηση καταλληλότητας φαίνεται από την πιο κάτω εξίσωση:

$$fitness = \# projects_satisfy_rule + \frac{1}{2 * std} + \frac{1}{length_of_rule}$$

Βελτίωση Καταλληλότητας (Fitness Improvement): Τα άτομα μπορούν να βελτιώσουν την καταλληλότητα τους είτε με βάση τη μέση τιμή της καταλληλότητας (mean fitness) του καλύτερου ατόμου είτε με βάση τη μέση τιμή της καταλληλότητας του πληθυσμού.

Υπολογισμός αριθμού απογόνων (Expected Number of Children): Είναι η μέθοδος με την οποία μπορούμε να υπολογίσουμε τον αριθμό των απογόνων κάθε

ατόμου. Η τεχνική του absolute ουσιαστικά δηλώνει ότι ο αριθμός των παιδιών κάθε ατόμου είναι ανάλογος της καταλληλότητας (fitness), ενώ οι μέθοδοι rank85 και rank89 δουλεύουν με βάση την ταξινόμηση κάθε ατόμου στον πληθυσμό.

Εμείς θα κάνουμε χρήση της πρώτης τεχνικής που έχουμε αναφέρει.

Γενετικοί Τελεστές (Genetic Operators): Οι γενετικοί τελεστές καθορίζουν την μέθοδο με την οποία θα δημιουργηθούν τα νέα μας άτομα:

- Διασταύρωση (Crossover): Τυχαίοι κόμβοι και από τους δύο γονείς αποκόβονται και τα δύο κλαδιά ανταλλάσσονται μεταξύ τους δημιουργώντας δύο νέους απογόνους.
- Μετάλλαξη (Mutation): Ένας τυχαίος κόμβος αποκόπτεται από ένα γονέα και αντικαθιστάται από ένα καινούργιο τυχαίο κόμβο.
- Μετάλλαξη Συρρίκνωσης (Shrink Mutation): Είναι η ίδια μέθοδος όπως και η μετάλλαξη με την διαφορά ότι προσδιορίζεται το μέγεθος του δένδρου που θα αποκοπεί.
- Μετάλλαξη Ανταλλαγής (Swap Mutation): Δύο τυχαία υπό δέντρα αποκόπτονται και ανταλλάζουν θέση μεταξύ τους.
- Μετάλλαξη Αντικατάστασης (Replace Mutation): Η συνάρτηση αυτή έχει δημιουργηθεί από εμάς και είναι μια διαφοροποίηση της συνάρτησης μετάλλαξης που παρέχεται από το εργαλείο. Η διαφορά εδώ είναι ότι αν το σημείο μετάλλαξης είναι τελεστής η συνάρτηση αυτή θα επιλέξει άλλο τελεστή να τον αντικαταστήσει, ενώ αν είναι τελεστέος θα αντικατασταθεί από ένα άλλο τελεστέο.
- Διασταύρωση & Μετάλλαξη (Cross & Mut): Και αυτή η συνάρτηση έχει δημιουργηθεί από εμάς και ουσιαστικά με αυτή θα έχουμε και διασταύρωση αλλά και μετάλλαξη, δηλαδή τα άτομα που θα δημιουργηθούν από την διασταύρωση θα περάσουν μετά από μετάλλαξη και μετά να τοποθετηθούν στον πληθυσμό για να περάσουν από την συνάρτηση καταλληλότητας.

Μέσα από εμπειρικά πειράματα που έχουμε εκτελέσει επιλέξαμε να χρησιμοποιήσουμε τους γενετικούς τελεστές Διασταύρωση, Μετάλλαξη Αντικατάστασης αλλά και την ταυτόχρονη εκτέλεση των δύο αυτών τελεστών. Για πιθανότητες επιλογής του κάθε ενός γενετικού τελεστή έχουμε καθορίσει να είναι 0.3 για τη διασταύρωση, 0.2 για τη μετάλλαξη αντικατάστασης, ενώ για τον συνδυασμό τους έχουμε θέσει πιθανότητα ίση του 0.6. Σε περίπτωση που θα γίνει χρήση κάποιας άλλης από τις επιτρεπτές τεχνικές μετάλλαξης η πιθανότητα που έχουμε θέσει για αυτές είναι 0.1.

4.3.2.2 Υποστηρικτικοί Παράμετροι Γενετικού Αλγορίθμου

Λειτουργίες (Functions): Η εφαρμογή αυτή υποστηρίζει τρία είδη λειτουργιών. Τους αριθμητικούς τελεστές, τους λογικούς τελεστές με βάση αριθμητικά δεδομένα και τους λογικούς τελεστές με βάση κατηγορικά δεδομένα. Οι αριθμητικοί τελεστές ουσιαστικά είναι μαθηματικές πράξεις της μορφής:

$$((X1+X2)^{X3})*(X3/X4 - \log(X2))$$

όπου X_i είναι παράγοντες

Από την άλλη, οι λογικοί τελεστές με αριθμητικά δεδομένα αποτελούν λογικές εκφράσεις της μορφής:

$$(X1 \leq 1.3) \ \&\& \ (X2 > 1.4)$$

όπου X_i είναι παράγοντες και οι τιμές είναι η ελάχιστη ή η μέγιστη τιμή της αντίστοιχης μεταβλητής.

Τέλος οι λογικοί τελεστές που παράγονται με χρήση κατηγορικών δεδομένων έχουν την μορφή:

$$(X1='A' \ || \ X1='B') \ \text{XOR} \ (X2='C')$$

όπου X_i είναι παράγοντες και 'A', 'B', 'C' πιθανές κατηγορικές τιμές της αντίστοιχης μεταβλητής.

Επιλογή Σημείου ή Διαστήματος (Point Estimate / Range Estimate): Αυτή η επιλογή σε περίπτωση που θα τρέξουμε αριθμητικούς τελεστές πάντα παίρνει τιμή την επιλογή σημείου, ενώ στις περιπτώσεις των λογικών και κατηγορικών εκτελέσεων ο χρήστης μπορεί να κάνει επιλογή μιας από τις δύο. Η επιλογή του χρήστη είναι ανάλογα με το αν επιθυμεί να κάνει πρόβλεψη με μορφή μιας τιμής ή πρόβλεψη με βάση κάποιο διάστημα. Στην περίπτωση της πρόβλεψης με μια τιμή έχουμε την αναλογία ευστοχίας (Hit Ratio), ενώ στην πρόβλεψη με βάση κάποιο διάστημα έχουμε την αναλογία επιτυχίας (Success Ratio).

- Αναλογία ευστοχίας είναι ο αριθμός των έργων (projects) που ικανοποιούν τον παραγόμενο κανόνα
- Αναλογία επιτυχίας είναι ο αριθμός των έργων τα οποία ικανοποιούν τον παραγόμενο κανόνα αλλά επίσης έχουν τιμή καταλληλότητας στο διάστημα: [mean-std effort ... mean+std effort].

Μέγεθος Δένδρου (Tree Size): Εδώ πρέπει να καθορίσουμε με βάση πιο κριτήριο θα μετρείται το μέγεθος του παραγόμενου δένδρου. Μετρήσεις γίνονται με βάση το βάθος του δένδρου (επίπεδα) ή με βάση των αριθμό των κόμβων του δένδρου.

Δυναμικό Επίπεδο (Dynamic Level): Καθορίζεται κατά πόσο επιτρέπεται στο μέγεθος του δένδρου να αλλάξει ή όχι και αν ναι αν μπορεί να μικρύνει.

Πολύ δυναμικό (Very Heavy): Παράλληλα με το πιο πάνω σε περίπτωση που θέλουμε να έχουμε δυναμικό επίπεδο μπορούμε να καθορίσουμε με αυτή την μεταβλητή κατά πόσο επιθυμούμε το μέγεθος όχι μόνο να αλλάξει αλλά και να μειωθεί σε μέγεθος, νοούμενου ότι ο παραγόμενος πληθυσμός είναι ο καλύτερος δυνατός.

Πόροι (Resources): Καθορισμός του πως θα κατανεμηθούν οι πόροι σε μια γενιά. Μπορεί να είναι σταθεροί ή να διατίθενται όταν χρειάζεται.

Είδος Δένδρου (Tree Pop Type): Καθορίζει την μορφή που θα έχει το παραγόμενο δένδρο, εντελώς ισορροπημένο, εντελώς ανισόροπο ή μίγμα των δύο.

Εκτελέσεις Πειράματος (Experiment Executions): Πόσες φορές (shuffles) θέλουμε να τρέξουμε το πείραμα με τις ίδιες ακριβώς επιλογές της οθόνης.

Όρια Μεγέθους Δένδρου (Tree Size Limitation): Πρέπει να καθοριστεί το μέγιστο μέγεθος του παραγόμενου δένδρου είτε αυτό είναι σε αριθμό κόμβων είτε είναι σε αριθμό επιπέδων.

Ποσοστό Δεδομένων για εκμάθηση (Training Set Percentage): Αυτό το ποσοστό αντιπροσωπεύει το ποσοστό των δεδομένων του αρχείου μας που θα χρησιμοποιηθεί για εκμάθηση. Το υπόλοιπο ποσοστό θα χρησιμοποιηθεί για επαλήθευση.

Ποσοστό ευστοχίας καταλληλότητας (Fitness Hits Percentage) & Ανοχή ευστοχίας καταλληλότητας (Fitness Hits Tolerance): Είναι δύο παράμετροι που χρησιμοποιούνται σαν κριτήρια τερματισμού εκτός από τον αριθμό γενιών που έχουν παραχθεί. Το αποτέλεσμα του καλύτερου ατόμου δεν πρέπει να είναι μικρότερο από ένα κάτω όριο αλλά ούτε και μεγαλύτερο από ένα άνω όριο. Για παράδειγμα αν δώσουμε τις τιμές [50 10] αντίστοιχα, αυτό σημαίνει ο αλγόριθμος θα τερματίσει αν το καλύτερο άτομο έχει τιμές $\pm 10\%$ του αναμενόμενου αποτελέσματος στο 50% των περιπτώσεων του.

Αποθήκευση Μεταβλητών (Save Variables): Κατά πόσο θέλουμε να φυλάξουμε τις τιμές των μεταβλητών του αλγορίθμου μας και αν ναι σε πόσο συχνό διάστημα να γίνεται αυτό. Για τους σκοπούς της παρούσας μελέτης καλύτερα να φυλάγουμε πάντα αυτές τις τιμές εφόσον θα τις χρειαστούμε στην συνέχεια για την δημιουργία γραφικών παραστάσεων.

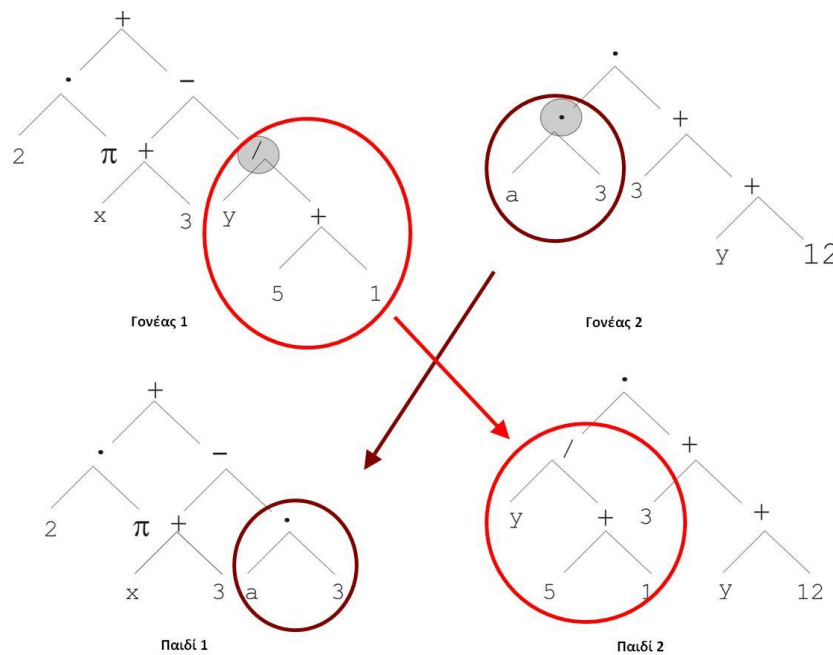
4.3.3 Εκτέλεση γενετικού προγραμματισμού

Μετά από την επιλογή όλων των παραμέτρων θα ξεκινήσει η εκτέλεση. Πρώτο βήμα είναι αφού διαβαστεί το αρχείο εισόδου, να χωριστεί σε δύο μέρη, στο σύνολο των δεδομένων εκμάθησης και στο σύνολο των δεδομένων ελέγχου. Ο διαχωρισμός αυτός γίνεται εντελώς τυχαία και με βάση το ποσοστό που έχει καθορίσει ο χρήστης.

Στη συνέχεια αφού έχουμε όλες τις απαραίτητες μεταβλητές, ξεκινά να εκτελείται ο γενετικός μας αλγόριθμος. Πρώτα πρέπει να δημιουργηθεί ο αρχικός μας πληθυσμός με βάση το μέγεθος που έχει καθορίσει ο χρήστης. Αυτός ο πληθυσμός θα δημιουργηθεί με εντελώς τυχαίο τρόπο και τα δένδρα που θα δημιουργηθούν για κάθε άτομο του πληθυσμού θα ακολουθούν το περιορισμό μεγέθους που έχει θέσει και πάλι ο χρήστης.

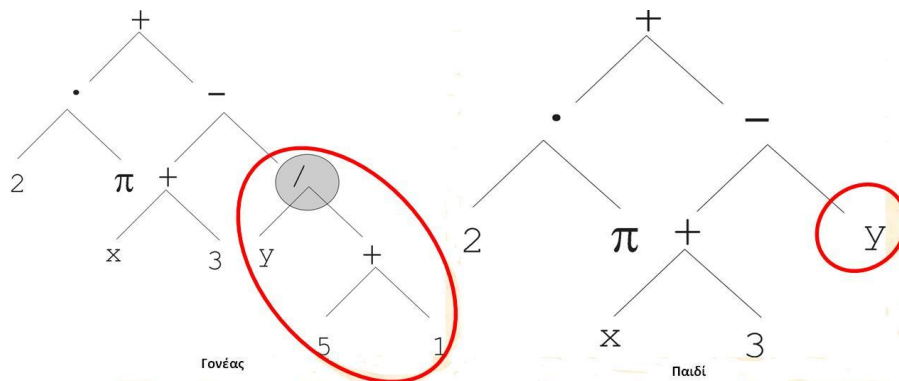
Αφού έχουμε δημιουργήσει τον αρχικό πληθυσμό μας, πρέπει να ξεκινήσουμε να κτίζουμε την πρώτη γενιά. Πρώτα θα επιλεγεί τυχαία από το σύνολο των διαθέσιμων τελεστών αλλά λαμβάνοντας υπόψη και τις πιθανότητες που έχουν καθοριστεί, ένας γενετικός τελεστής. Για να είναι αυτό εφικτό πρέπει πρώτα να γίνει η διαδικασία της δειγματοληψίας, όπου με βάση την επιλεγμένη μέθοδο θα γίνει επιλογή των γονέων του υπό δημιουργία ατόμου από τον πληθυσμό που έχουμε ήδη δημιουργήσει. Ο αριθμός των γονέων που χρειάζονται εξαρτάται από τον γενετικό τελεστή που έχει επιλεγεί. Για παράδειγμα ο γενετικός τελεστής της διασταύρωσης χρειάζεται δύο γονείς ενώ ο γενετικός τελεστής χρειάζεται μόνο ένα γονέα. Αφού λοιπόν γίνει η επιλογή του γονέα ή των γονέων θα γίνει εφαρμογή του γενετικού τελεστή.

Ο γενετικός τελεστής της διασταύρωσης παίρνει δύο γονείς και επιλέγει τυχαία από τον κάθε γονέα ένα υπό-δένδρο και τα ανταλλάζει μεταξύ τους. Η διαδικασία της διασταύρωσης φαίνεται και σχηματικά στην πιο κάτω εικόνα (εικόνα 4.6).



Εικόνα 4.6: Διαδικασία τελεστή διασταύρωσης [27]

Ο γενετικός τελεστής της μετάλλαξης στην πιο κοινή της μορφή παίρνει ένα άτομο (γονέα) επιλέγει τυχαία ένα υπό-δένδρο του και το αντικαθιστά με ένα άλλο τυχαίο υπό-δένδρο. Η διαδικασία αυτή φαίνεται και σχηματικά στο πιο κάτω σχήμα (εικόνα 4.7).



Εικόνα 4.7: Διαδικασία τελεστή μετάλλαξης [27]

Εκτός από την κοινή αυτή μορφή του τελεστή της μετάλλαξης, το προτεινόμενο σύστημα παρέχει και την μετάλλαξη αντικατάστασης. Η διαφορά στην λειτουργία αυτού του τελεστή είναι στο γεγονός ότι αντί να έχουμε αντικατάσταση με τυχαίο δένδρο έχουμε αντικατάσταση αναλόγως του τι έχουμε στο σημείο μετάλλαξης του γονέα. Αν στο σημείο μετάλλαξης έχουμε τελεστή τότε θα επιλεγεί ένας άλλος τελεστής για να τον αντικαταστήσει, ενώ αν είναι τελεστέος τότε θα επιλεγεί τυχαία ένας άλλος τελεστέος.

Και με τα δύο αυτά είδη τελεστών τα νέα παραγόμενα δένδρα (παιδιά) μπορεί να έχουν μεγαλύτερο μέγεθος από τον γονέα τους.

Η διαδικασία της επιλογής τελεστή, επιλογής γονέων και της εφαρμογής του τελεστή επαναλαμβάνεται μέχρι να δημιουργήσουμε ένα νέο πληθυσμό ο οποίος να έχει το μέγεθος που έχει καθορίσει ο χρήστης. Αφού γίνει αυτό έχουμε ένα καινούργιο πληθυσμό και πρέπει να υπολογίσουμε για κάθε άτομο του πληθυσμού αυτού ποια είναι η τιμή καταλληλότητας του. Αυτό θα γίνει με βάση την επιλεγμένη μέθοδο από το χρήστη πάνω στα δεδομένα εκμάθησης. Τα επόμενα βήματα είναι αυτά του ελιτισμού αλλά και της επιβίωσης. Με βάση την τιμή της καταλληλότητας κάθε ατόμου αλλά και την μέθοδο ελιτισμού που έχει επιλέξει ο χρήστης θα δοθούν προτεραιότητας σε κάθε ένα άτομο του νέου πληθυσμού. Με την επιβίωση καθορίζουμε πόσα άτομα θα κρατήσουμε στον πληθυσμό μας. Το άτομο από τον πληθυσμό το οποίο έχει την καλύτερη τιμή καταλληλότητας θα θεωρηθεί και θα αποθηκευτεί ότι είναι το καλύτερο.

Τα πιο πάνω βήματα (μετά από την αρχικοποίηση του πληθυσμού) θα επαναληφθούν τόσες φορές όσες και ο αριθμός των γενιών που επέλεξε ο χρήστης ή μέχρι κάποιο άλλο κριτήριο τερματισμού ευσταθεί. Όταν τελειώσουν όλα αυτά τα βήματα και καταλήξουμε σε ένα άτομο με την καλύτερη τιμή καταλληλότητας τότε θεωρούμε ότι το άτομο αυτό είναι και η λύση της εκτέλεσης μας και επομένως μπορούμε να κάνουμε αποτίμηση της παραγόμενης εξίσωσης στα δεδομένα ελέγχου για να επαληθεύσουμε αν το σύστημα μας κατάφερε να δουλέψει σωστά.

Όταν έχει τελειώσει η εκτέλεση που περιγράψαμε, το σύστημα θα δημιουργήσει το αρχείο εξόδου. Για να γίνει αυτό θα πρέπει πρώτα να γίνουν κάποιοι υπολογισμοί ανάλογα με το είδος της εκτέλεσης. Αν είμαστε σε αριθμητική εκτέλεση, τότε θα πρέπει να υπολογιστεί η τιμή καταλληλότητας της επιλεγμένης εξίσωσης με βάση τα δεδομένα ελέγχου καθώς επίσης και όλοι οι συντελεστές λάθους τόσο στο σύνολο των δεδομένων εκμάθησης όσο και στο σύνολο των δεδομένων ελέγχου (βλέπε 4.3.4). Αν η εκτέλεση μας είναι λογική ή κατηγορική τότε θα υπολογίσουμε τη μέση τιμή προσπάθειας και τις αντίστοιχες αναλογίες που θέλουμε να εκτυπώσουμε στο αρχείο εξόδου (βλέπε 4.3.4).

Τα πιο πάνω αποτελούν τα βήματα που θα ακολουθηθούν σε μια εκτέλεση. Ανάλογα με τον αριθμό των εκτελέσεων που έχει επιλέξει ο χρήστης θα επαναληφθεί η πιο πάνω διαδικασία και θα έχουμε σαν αποτέλεσμα στο αρχείο εξόδου τόσες εξισώσεις όσες και οι επαναλήψεις που έχουν γίνει. Σε κάθε επανάληψη το σύνολο των δεδομένων εκμάθησης και των δεδομένων ελέγχου είναι διαφορετικό αφού όπως έχουμε πει ο διαχωρισμός είναι εντελώς τυχαίος. Επίσης τυχαίος και επομένως διαφορετικός θα είναι και ο αρχικός πληθυσμός σε κάθε εκτέλεση. Το μόνο κοινό όλων των εκτελέσεων που θα γίνουν θα είναι οι παράμετροι που έχει επιλέξει ο χρήστης.

4.3.4 Δημιουργία αρχείου εξόδου

Το τελευταίο στάδιο κάθε εκτέλεσης είναι η παραγωγή ενός αρχείου εξόδου. Το αρχείο αυτό έχει την μορφή ενός αρχείου κειμένου (txt). Ανάλογα με την εκτέλεση έχουμε διαφορετικό αρχείο εξόδου. Για τις λογικές και κατηγορικές εκτελέσεις η μορφή του αρχείου εξόδου είναι η ίδια.

Στις αριθμητικές εκτελέσεις αποθηκεύουμε στο αρχείο την αριθμητική εξίσωση του καλύτερου ατόμου, την τιμή της καταλληλότητας του καλύτερου ατόμου, την ολική τιμή καταλληλότητας της παραγόμενης εξίσωσης καθώς επίσης και τα πιο κάτω λάθη:

- Κανονικοποιημένη Ρίζα Μέσου Τετραγωνικού Σφάλματος (Normalized Root Mean Squared Error): Βασικό στοιχείο αυτής της μεθόδου αξιολόγησης είναι το ότι ανιχνεύει την ποιότητα των προβλέψεων επικεντρώνοντας καθαρά στην σύγκρισή της με αυτήν ενός κινητού μέσου. Το ιδανικότερο είναι να πάρουμε πρόβλεψη όπου η κανονικοποιημένη ρίζα μέσου τετραγωνικού σφάλματος να είναι 0, δηλαδή έχουμε 100% επιτυχία. Αν η απόκλιση των αυθεντικών τιμών των δειγμάτων είναι πολύ μικρός αριθμός ή αν τα δείγματα είναι τιμές πολύ κοντά στο μέσο της σειράς, με πολύ μικρές διακυμάνσεις τότε αυτό το κριτήριο θα δώσει πολύ μεγάλες τιμές που αυτό μπορούμε να πούμε είναι και το μειονέκτημα του κριτηρίου αυτού.

Για τον υπολογισμό αυτού του σφάλματος χρησιμοποιούμε τη Ρίζα Μέσου Τετραγωνικού Σφάλματος

$$RMSE(n) = \sqrt{\frac{1}{n} \sum_{i=1}^n [x_{pred}(i) - x_{act}(i)]^2}$$

Ενώ η Κανονικοποιημένη ρίζα είναι

$$NRMSE(n) = \frac{RMSE(n)}{\sigma_{\Delta}} = \frac{RMSE(n)}{\sqrt{\frac{1}{n} \sum_{i=1}^n [x_{act}(i) - \bar{x}_n]^2}}$$

- Συντελεστής Συσχέτισης (Correlation Coefficient): Με αυτό το κριτήριο μπορούμε να δούμε κατά πόσο οι προβλεπόμενες τιμές ακολουθούν αυξητική ή μειωτική εναλλαγή σε σχέση με τις πραγματικές τιμές. Σε αντίθεση με το προηγούμενο κριτήριο, σε αυτό το κριτήριο ιδανικό θα ήταν αν οι τιμές που θα μας δώσει είναι κοντά στο ένα (είτε κοντά στο -1 είτε κοντά στο +1). Σε αυτή την περίπτωση οι προβλέψεις μιμούνται την κίνηση, την τάση των πραγματικών τιμών σε κλίση. Οι αρνητικές τιμές του Συντελεστή Συσχέτισης ερμηνεύονται σαν μια καθρεφτική αντίδραση των πραγματικών και των προβλεπόμενων τιμών, δηλαδή εκεί που

υπάρχει αυξητική τάση στην πραγματική σειρά, υπάρχει μειωτική τάση στην προβλεπόμενη σειρά και το αντίστροφο. Ενώ όταν η τιμή του Συντελεστή Συσχέτισης είναι κοντά στο μηδέν, τότε έχουμε ανικανότητα της μεθόδου να παράγει προβλέψεις που να αναπαράγουν την κίνηση της αυθεντικής σειράς, των πραγματικών τιμών.

$$CC = \frac{\sum_{i=1}^n [(x_{act}(i) - \bar{x}_{act,n}) * (x_{pred}(i) - \bar{x}_{pred,n})]}{\sqrt{\left[\sum_{i=1}^n (x_{act}(i) - \bar{x}_{act,n})^2 \right] * \left[\sum_{i=1}^n (x_{pred}(i) - \bar{x}_{pred,n})^2 \right]}}$$

Όπου $\bar{x}_{act,n}$ είναι οι μέσες τιμές των n πραγματικών δειγμάτων και $\bar{x}_{pred,n}$ οι μέσες τιμές των προβλεπόμενων δειγμάτων.

- Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error): Το κριτήριο σφάλματος αυτό ουσιαστικά δίνει το σφάλμα των προβλέψεων εκφρασμένο ως την μέση τιμή της τετραγωνικής απόκλισης των πραγματικών τιμών από τις προβλεπόμενες τιμές. Το λάθος υπολογίζεται από την τιμή της διαφοράς ανάμεσα στις τιμές των δειγμάτων μας. Το κριτήριο αυτό είναι εξαρτώμενο από την κλίμακα των δεδομένων, αφού το λάθος υπολογίζεται από την τιμή της διαφοράς ανάμεσα στις τιμές των δειγμάτων.

$$MSE(n) = \frac{1}{n} \sum_{i=1}^n [x_{act}(i) - x_{pred}(i)]^2$$

Όπου n, ο αριθμός των προβλέψεων, $x_{act}(i)$ η αυθεντική πραγματική τιμή του δείγματος i και $x_{pred}(i)$ η προβλεπόμενη τιμή του δείγματος i.

- Μέσο Σφάλμα Συσχέτισης (Mean Relative Error): Το κριτήριο αυτό, δίνει το σφάλμα των προβλέψεων εκφρασμένο ως την απόλυτη τιμή της απόκλισης των πραγματικών τιμών από τις προβλεπόμενες τιμές.

$$MRE(n) = \frac{\frac{1}{n} \sum_{i=1}^n |x_{act}(i) - x_{pred}(i)|}{x_{act}(i)}$$

- Μέσο Απόλυτο Σφάλμα (Mean Absolute Error): Το κριτήριο αυτό δίνει το σφάλμα των προβλέψεων εκφρασμένο ως την απόλυτη τιμή της απόκλισης των πραγματικών τιμών από τις προβλεπόμενες τιμές. Και αυτό το λάθος είναι εξαρτώμενο από την κλίμακα των δεδομένων, σε μικρότερη όμως ευαισθησία αφού το λάθος υπολογίζεται από την απόλυτη τιμή της διαφοράς ανάμεσα στις τιμές των δειγμάτων.

$$MAE(n) = \frac{1}{n} \sum_{i=1}^n |x_{act}(i) - x_{pred}(i)|$$

Όπου n , ο αριθμός των προβλέψεων, $x_{act}(i)$ η αυθεντική πραγματική τιμή του δείγματος i και $x_{pred}(i)$ η προβλεπόμενη τιμή του δείγματος i .

- Πρόβλεψη (Pred): Το κριτήριο αυτό δίνει τον αριθμό των προβλέψεων οι οποίες έχουν σωστή πρόβλεψη σε σχέση με τον ολικό αριθμό των δεδομένων μας.

$$pred(n) = \frac{k}{n}$$

Για να καθορίσουμε αν η πρόβλεψη έχει γίνει σωστά χρησιμοποιούμε την ακόλουθη εξίσωση:

$$RE(n) = \frac{|x_{act}(i) - x_{pred}(i)|}{x_{act}(i)}$$

Σε αυτή την πειραματική διαδικασία θα έχουμε αυτό το λάθος για το 25% , επομένως αυτό που υπολογίζουμε εδώ είναι πόσες γραμμές έχουν τιμή RE πιο μικρή από το επίπεδο 25%.

- Ισοζυγισμένο Σχετικό Λάθος (Balanced Relative Error): Μετρούμε την ακρίβεια πρόβλεψης ανάλογα αν η προβλεπόμενη τιμή είναι μικρότερη ή μεγαλύτερη από την πραγματική.

$$BRE(n) = \sum_{i=1}^n \left\{ \begin{array}{l} \frac{x_{act}(i) - x_{pred}(i)}{x_{act}(i)}, \text{ if } x_{act}(i) \leq x_{pred}(i) \\ \frac{x_{act}(i) - x_{pred}(i)}{x_{pred}(i)}, \text{ if } x_{act}(i) > x_{pred}(i) \end{array} \right\}$$

Τα πιο πάνω αναφερθέντα λάθη θα παραχθούν τόσο για τα δεδομένα εκμάθησης, όσο και για τα δεδομένα ελέγχου. Για την αποτίμηση των αποτελεσμάτων θα δώσουμε ιδιαίτερη έμφαση σε τρία κριτήρια αξιολόγησης του λάθους, στο μέσο σφάλμα συσχέτισης, στο συντελεστή συσχέτισης και στην κανονικοποιημένη ρίζα μέσου τετραγωνικού σφάλματος. Για να θεωρήσουμε μια εξίσωση ότι είναι καλή θα πρέπει οι τιμές για το μέσο σφάλμα συσχέτισης και την κανονικοποιημένη ρίζα μέσου τετραγωνικού σφάλματος των δεδομένων εκμάθησης να είναι μικρότερες από τις αντίστοιχες τιμές του συνόλου των δεδομένων ελέγχου καθώς επίσης και η τιμή του συντελεστή συσχέτισης των δεδομένων εκμάθησης να είναι μεγαλύτερη από την αντίστοιχη τιμή των δεδομένων ελέγχου. Επιπρόσθετα με αυτό, κριτήριο θα αποτελέσουν και οι τιμές αυτών των τριών συντελεστών στα δεδομένα ελέγχου. Ποιο συγκεκριμένα το μέσο σφάλμα συσχέτισης πρέπει να είναι μικρότερο από 0.6, ο συντελεστής συσχέτισης πρέπει να είναι μεγαλύτερος από 0.85 και τέλος η κανονικοποιημένη ρίζα τετραγωνικού σφάλματος να είναι μικρότερη από 0.5.

Αναφορικά με τις εκτελέσεις με λογικές εκφράσεις είτε αυτές είναι με βάση αριθμητικά δεδομένα είτε αυτές είναι με κατηγορικά δεδομένα το παραγόμενο αρχείο λαθών είναι διαφορετικό και περιέχει την λογική εξίσωση και τα πιο κάτω στοιχεία:

- Μέση τιμή της προσπάθειας για τα δεδομένα εκμάθησης (Mean effort). Για να πάρουμε αυτή τη μέση τιμή βρίσκουμε ποιες από τις γραμμές των δεδομένων μας ικανοποιούν της συγκεκριμένη εξίσωση και παίρνουμε την πραγματική τιμή καταλληλότητας τους. Με αυτές τις τιμές θα πάρουμε την μέση τιμή και αυτή θεωρείται ως η τιμή καταλληλότητας της εξίσωσης αυτής.

$$\text{Effort}_{\text{pred}} = \text{mean}(\text{effort}_{\text{act}})$$

- Σταθερή απόκλιση της καταλληλότητας για τα δεδομένα εκμάθησης (Standard Deviation). Με τον ίδιο τρόπο που αναφέραμε πιο πάνω βρίσκουμε την τιμή κάθε γραμμής εισόδου που ικανοποιεί την εξίσωση και αντί να πάρουμε την μέση τιμή όπως προηγουμένως παίρνουμε το standard deviation των τιμών αυτών.

$$\text{Std}_{\text{effort}} = \text{std}(\text{effort}_{\text{act}})$$

- Αναλογία Ευστοχίας των δεδομένων εκμάθησης (Hit Ratio Training): Είναι ο αριθμός των γραμμών εισόδου που ικανοποιούν την παραγόμενη εξίσωση.
- Αναλογία Ευστοχίας των δεδομένων ελέγχου (Hit Ratio Testing)
- Αναλογία Επιτυχίας των δεδομένων ελέγχου (Success Ratio): Είναι ο αριθμός των γραμμών εισόδου που ικανοποιούν την παραγόμενη εξίσωση αλλά επίσης έχουν τιμή καταλληλότητας στο διάστημα:

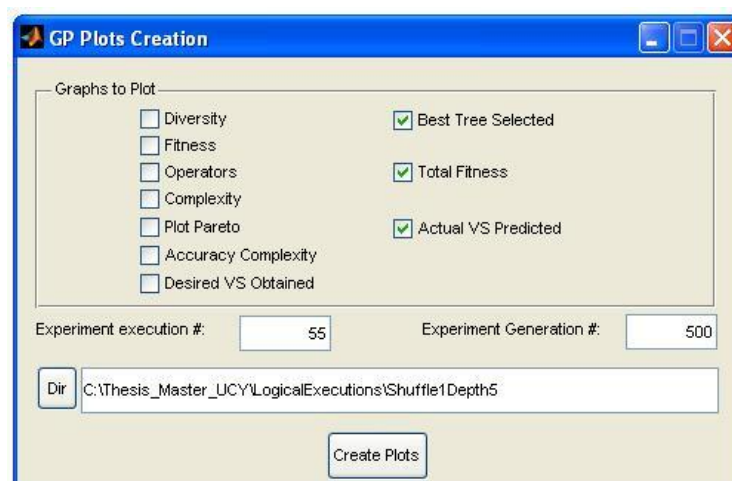
$$[\text{Effort}_{\text{pred}} - \text{Std}_{\text{effort}} \quad \dots \quad \text{Effort}_{\text{pred}} + \text{Std}_{\text{effort}}]$$

Όπου η μέση τιμή και η σταθερή απόκλιση που αναφέρουμε εδώ είναι οι τιμές που παίρνουμε με βάση τα δεδομένα εκμάθησης.

4.3.5 Παραγόμενες Γραφικές Παραστάσεις

Οι γραφικές παραστάσεις δεν έχουν συμπεριληφθεί στο σύστημα αλλά είναι ένα ανεξάρτητο κομμάτι το οποίο μπορεί να εκτελεστεί μετά από κάποια εκτέλεση υπό την προϋπόθεση όμως ότι στην εκτέλεση που έχουμε κάνει έχουμε επιλέξει να φυλάξουμε απαραίτητα τις τιμές των μεταβλητών μας. Αυτό είναι εφικτό με την επιλογή “Save Variables” του προτεινόμενου συστήματος.

Η διεπαφή των γραφικών παραστάσεων είναι απλή και φαίνεται στην εικόνα 4.8 πιο κάτω.



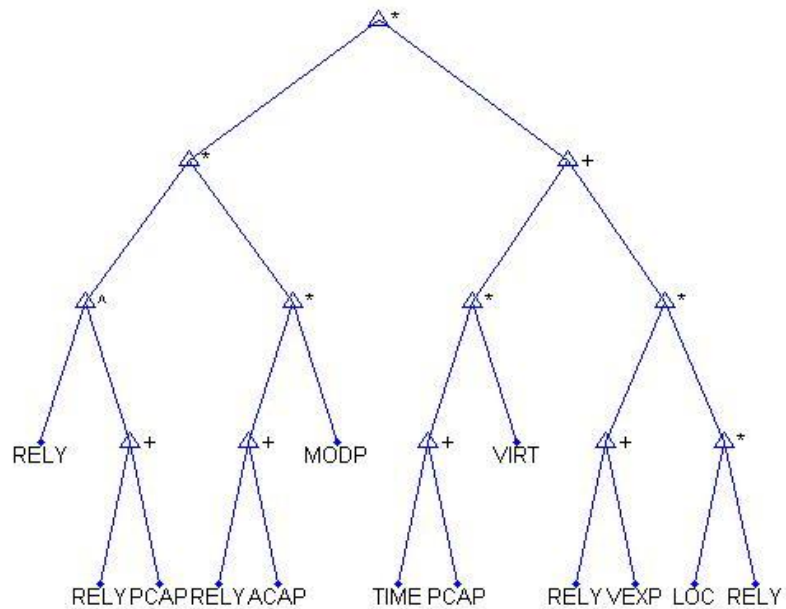
Εικόνα 4.8: Οθόνη επιλογής γραφικών παραστάσεων

Προκειμένου να μπορέσουμε να δημιουργήσουμε οποιαδήποτε από τις γραφικές παραστάσεις, θα πρέπει να γνωρίζουμε τον αριθμό του πειράματος αλλά και την γενιά για την οποία θέλουμε να δούμε γραφικές απεικονίσεις, αφού τα ονόματα των αρχείων που φυλάγονται τα δεδομένα μας κατά την διάρκεια των πειραμάτων παίρνουν ονομασίες με βάση αυτές τις τιμές. Για παράδειγμα αν έχουμε εκτελέσει πείραμα το οποίο αποτελείται από 500 γενιές και 100 επαναλήψεις και θέλουμε να δούμε γραφικές παραστάσεις για την επανάληψη 55. Το σύστημα κατά την εκτέλεση του γενετικού προγραμματισμού θα δημιουργήσει για κάθε επανάληψη ένα φάκελο (folder) που θα ονομάζεται "SaveVarsVals" και τον αριθμό της επανάληψης (στην προκειμένη περίπτωση θα δημιουργηθεί ο φάκελος "SaveVarsVals55") και εκεί θα αποθηκεύεται σε αρχείο με όνομα τον αριθμό της γενιάς (στο παράδειγμα μας θα δημιουργηθεί το αρχείο "55.mat") όλες οι πληροφορίες. Με την ίδια λογική που το προτεινόμενο σύστημα θα αποθηκεύσει τις απαραίτητες τιμές, το προτεινόμενο σύστημα των γραφικών παραστάσεων θα αναζητήσει τα απαραίτητα αρχεία προκειμένου να τα ανοίξει και να έχει όλες τις πληροφορίες που χρειάζεται για την δημιουργία των επιλεγμένων παραστάσεων.

Στην οθόνη που βλέπουμε πιο πάνω παρέχονται διάφορες γραφικές παραστάσεις. Στο αριστερό μέρος βρίσκονται οι γραφικές παραστάσεις που μπορεί να δώσει το εργαλείο GPLab ενώ στο δεξιό μέρος είναι οι γραφικές παραστάσεις που έχουμε δημιουργήσει εμείς προκειμένου να αναλύσουμε τα αποτελέσματα που έχουμε πάρει. Στην ανάλυση μας δεν έγινε χρήση καμιάς από τις διαθέσιμες γραφικές αλλά βασιστήκαμε μόνο σε αυτές που έχουμε δημιουργήσει. Ας δούμε πιο αναλυτικά τις γραφικές παραστάσεις που έχουμε χρησιμοποιήσει για τους σκοπούς αυτής της μελέτης.

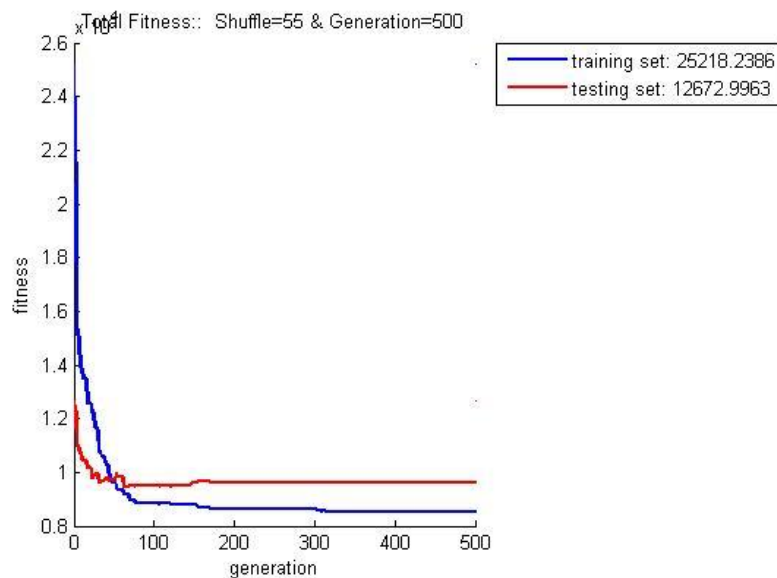
Γραφική Παράσταση Καλύτερου Δέντρου (Best Tree): Με αυτή την παράσταση γίνεται μια γραφική απεικόνιση του καλύτερου δέντρου που έχει δημιουργηθεί στην επιλεγμένη γενιά (Εικόνα 4.9). Η εξίσωση που αντιστοιχεί σε αυτή την αναπαράσταση είναι:

$$((RELY^{(RELY+PCAP)})*((RELY+ACAP)*MODP)*(((TIME+PCAP)*VIRT))+((RELY+VEXP)*(LOC*RELY))$$



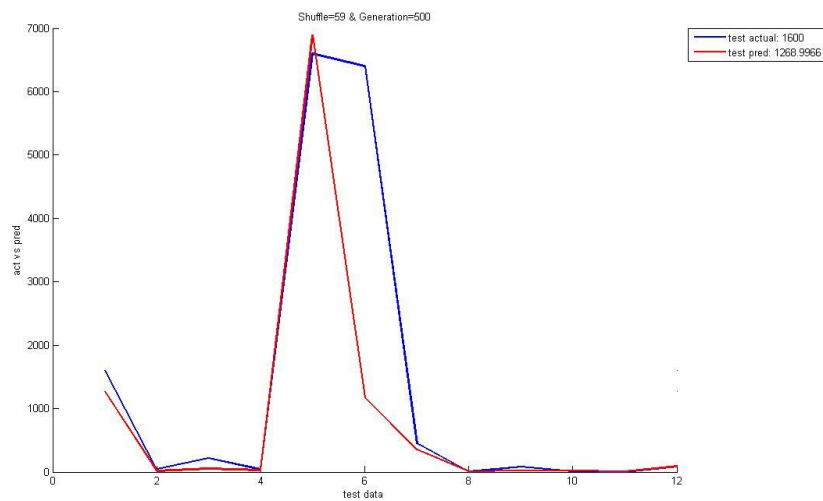
Εικόνα 4.9: Γραφική Απεικόνιση Δέντρου

Γραφική Παράσταση Συνάρτησης Καταλληλότητας (Total Fitness): Με αυτή την παράσταση δείχνουμε την εξέλιξη της τιμής της ολικής καταλληλότητας (total fitness) σε κάθε γενιά (Εικόνα 4.10). Με μπλε χρώμα μπορούμε να δούμε την τιμή αυτή για τα δεδομένα εκμάθησης μας, ενώ με κόκκινο χρώμα μπορούμε να δούμε την τιμή αυτή για τα δεδομένα ελέγχου μας.



Εικόνα 4.10: Γραφική Απεικόνιση Total Fitness

Γραφική Παράσταση Πραγματικού - Υπολογιζόμενου (Actual VS Predicted): Με αυτή την παράσταση (Εικόνα 4.11) μπορούμε να δούμε και να συγκρίνουμε για κάθε ένα από τους τελεστές μας, την πραγματική τιμή ικανότητας αλλά και την υπολογιζόμενη τιμή ικανότητας με βάση την συνάρτηση καταλληλότητας. Έτσι θα μπορέσουμε να δούμε κατά πόσο έχουμε φτάσει σε μια λύση που να είναι όσο το δυνατό πιο κοντά στην πραγματικότητα και επομένως αν το σύστημα μας δουλεύει σωστά.



Εικόνα 4.11: Γραφική Απεικόνιση Πραγματικού - Υπολογιζόμενου

Κεφάλαιο 5

Πειραματική Διαδικασία

- 5.1 Εισαγωγή
 - 5.2 Γενική Ανασκόπηση Πειραμάτων
 - 5.2.1 Πειραματικές Εκτελέσεις Αριθμητικών Συναρτήσεων
 - 5.2.2 Πειραματικές Εκτελέσεις Λογικών Συναρτήσεων
 - 5.2.3 Πειραματικές Εκτελέσεις Κατηγορικών Συναρτήσεων
 - 5.3 Παρουσίαση Αποτελεσμάτων
 - 5.3.1 Αποτελέσματα Αριθμητικών Εκτελέσεων
 - 5.3.2 Αποτελέσματα Λογικών Εκτελέσεων
 - 5.3.3 Αποτελέσματα Κατηγορικών Εκτελέσεων
 - 5.4 Συμπεράσματα και Γραφικές Παραστάσεις
-

5.1 Εισαγωγή

Η ακριβής αλλά ταυτόχρονα και αξιόπιστη πρόβλεψη του κόστους ανάπτυξης λογισμικού είναι αρκετά πολύπλοκη αλλά και προκλητική διαδικασία. Έχοντας υπόψη μας αυτό και με την βοήθεια του συστήματος που έχουμε αναφέρει πιο πάνω, προσπαθήσαμε μέσω μιας σειράς πειραμάτων να εξάγουμε όσο το δυνατό πιο εύρωστους κανόνες οι οποίοι αναμένουμε να βοηθήσουν ιδιαίτερα την πρόβλεψη του κόστους.

Η πειραματική διαδικασία αποτελείται από δύο στάδια. Στο πρώτο στάδιο έγιναν κάποιες μικρές εκτελέσεις με διαφορετικές παραμέτρους έτσι ώστε να καταφέρουμε να βρούμε ποια είναι η τιμή κάθε μιας από τις παραμέτρους εισόδου η οποία μπορεί να μας δώσει τα καλύτερα δυνατά αποτελέσματα. Με βάση τα αποτελέσματα αυτά προχωρήσαμε στις μεγάλες εκτελέσεις των πειραμάτων.

5.2 Γενική Ανασκόπηση Πειραμάτων

Για κάθε μια από τις παραμέτρους εισόδου ο χρήστης μπορεί να επιλέξει περισσότερες από μια τιμές. Επομένως το πρώτο δίλημμα που είχαμε να αντιμετωπίσουμε είναι ποια τιμή για κάθε μια παράμετρο είναι η καλύτερη δυνατή. Θεωρητικά όλες οι επιλογές μπορούν να δουλέψουν σωστά και να δώσουν σχετικά καλά αποτελέσματα. Πρακτικά η εκτέλεση αντιπροσωπευτικών πειραμάτων με όλους τους συνδυασμούς τιμών παραμέτρων δεν είναι εφικτή. Μια μικρή προεργασία πρέπει να γίνει έτσι ώστε να μπορέσουμε να ανιχνεύσουμε τον καλύτερο δυνατό συνδυασμό με στόχο πάντα τα πιο ακριβή αποτελέσματα. Η προεργασία αυτή είναι μικρές σε μέγεθος εκτελέσεις με πραγματικά όμως δεδομένα και σύγκριση των αποτελεσμάτων κάθε εκτέλεσης. Μέσα από την εμπειρική αυτή διαδικασία που έχουμε κάνει, έχουμε καταλήξει ότι για να πάρουμε τα καλύτερα δυνατά αποτελέσματα θα πρέπει έχουμε τις ακόλουθες τιμές σε κάποιες από τις παραμέτρους εισόδου:

A/A	Παράμετρος	Τιμή
1	Μέγεθος Πληθυσμού	100
2	Ελιτισμός	Ολικός Ελιτισμός
3	Επιβίωση	Πηγές
4	Δειγματοληψία	Διπλό Τουρνουά
5	Συνάρτηση Καταλληλότητας	RegFitness για αριθμητικές LogFitness για λογικές & κατηγορικές
6	Βελτίωση Καταλληλότητας	Μέση τιμή καλύτερης εκτέλεσης
7	Αριθμός Παιδιών	Απόλυτος
8	Γενετικοί Τελεστές	Διασταύρωση, Μετάλλαξη Αντικατάσταση, Διασταύρωση & Μετάλλαξη Αντικατάσταση
9	Εκτελέσεις Πειραμάτων	100
10	Περιορισμός Μεγέθους Δένδρου	3, 4, 5, 6 και 7 για βάθος 16, 20 και 28 για αριθμό κόμβων
11	Ποσοστό Ευστοχίας	60%
12	Ανοχή Ευστοχίας	10%
13	Τύπος Παραγόμενων Δένδρων	Ισορροπημένα και μη ισορροπημένα δένδρα
14	Δυναμικό Επίπεδο	Ναι για βάθος Πολύ δυναμικό για κόμβους
15	Αποθήκευση Τιμών Μεταβλητών	Πάντα
16	Πολύ Δυναμικό	Όχι
17	Πηγές	Στατικές

Εικόνα 5.1: Τιμές Παραμέτρων Πειραμάτων

Η πειραματική διαδικασία θα έχει ως βάση τις πιο πάνω επιλεγμένες τιμές. Τα πειράματα που θα εκτελέσουμε θα γίνουν τόσο με βάση το μέγιστο βάθος που θέλουμε να έχει το παραγόμενο αποτέλεσμα, αλλά και με βάση το μέγιστο αριθμό κόμβων του

παραγόμενου δένδρου. Μπορούμε να διαχωρίσουμε τις εκτελέσεις μας με βάση το είδος των παραγόμενων συναρτήσεων:

1. Αριθμητικές Συναρτήσεις (Arithmetic)
2. Λογικές Συναρτήσεις με αριθμητικούς τελεστές (Logical)
3. Λογικές Συναρτήσεις με κατηγορικούς τελεστές (Categorical)

5.2.1 Πειραματικές Εκτελέσεις Αριθμητικών Συναρτήσεων

Ένα από τα βασικότερα στοιχεία των συναρτήσεων μας αποτελούν οι τελεστές και οι τελεστέοι. Στις αριθμητικές συναρτήσεις το προτεινόμενο σύστημα θεωρεί ως τελεστέο το κάθε χαρακτηριστικό (στήλη) που υπάρχει στο αρχείο δεδομένων μας. Από την άλλη το σύνολο των διαθέσιμων τελεστών είναι σταθερό και αποτελείται από τις εξής μαθηματικές πράξεις:

- Πρόσθεση (+)
- Αφαίρεση (-)
- Πολλαπλασιασμός (*)
- Διαίρεση (/)
- Δύναμη (^)
- Λογάριθμος - Φυσικός Λογάριθμος, Λογάριθμος με βάση το 2 και Λογάριθμος με βάση το 10 (\log , \log_2 , \log_{10})

Για τις αριθμητικές συναρτήσεις έχουμε κάνει διάφορες μικρές εκτελέσεις προκειμένου να αποφασίσουμε ποιες είναι οι καλύτερες τιμές για κάποιες παραμέτρους έτσι ώστε να έχουμε τα καλύτερα αποτελέσματα. Η μια ομάδα εκτελέσεων αφορούσε την συνάρτηση καταλληλότητας ενώ η δεύτερη αφορούσε τους διάφορους συνδυασμούς μεταξύ των εκτελέσεων με βάση το βάθος ή τον αριθμό των κόμβων του παραγόμενου δένδρου. Όσον αφορά την πρώτη ομάδα εκτελέσεων είχαμε θέσει σταθερές όλες τις παραμέτρους εισόδου και αλλάξαμε μονάχα την συνάρτηση καταλληλότητας. Η πρώτη συνάρτηση

καταλληλότητας που δοκιμάσαμε είναι η συνάρτηση που θεωρούσε σαν τιμή ικανότητας μιας εξίσωσης την τιμή του μέσου σφάλματος συσχέτισης MRE (βλέπε 4.3.3), ενώ η δεύτερη συνάρτηση καταλληλότητας αφορούσε την απόλυτη διαφορά μεταξύ της πραγματικής και της υπολογιζόμενης τιμής προσπάθειας (βλέπε 4.3.2). Τα αποτελέσματα που πήραμε σε όλες τις περιπτώσεις έδειχναν ότι καλύτερα αποτελέσματα παίρναμε με την δεύτερη συνάρτηση (regfitness). Αυτό έγινε και στις περιπτώσεις όπου οι εκτελέσεις βασίζονταν στο βάθος αλλά και στις περιπτώσεις που βασίζονταν στον αριθμό των κόμβων του δένδρου. Αυτός ήταν και ο λόγος που μας ώθησε να επιλέξουμε να κάνουμε όλες τις εκτελέσεις μας με την δεύτερη συνάρτηση. Για αυτή την συνάρτηση καταλληλότητας θέλουμε να πάρουμε τις λύσεις οι οποίες έχουν όσο το δυνατό χαμηλότερη τιμή.

Αρχικά είχαμε κάνει κάποιες εκτελέσεις οι οποίες είχαν μέγεθος συνόλου δεδομένων εκμάθησης 80% και 20% μέγεθος των δεδομένων ελέγχου. Αριθμό γενιών κρατήσαμε το 500. Στα αποτελέσματα που είχαμε πάρει παρατηρήσαμε ότι είχαμε μεγάλο πλήθος αποτελεσμάτων στα οποία οι τιμές των συντελεστών λάθους των δεδομένων εκμάθησης δεν ήταν καλύτερες από τις αντίστοιχες τιμές των δεδομένων ελέγχου. Αυτό μας έδειχνε ότι το σύστημα μας δεν μπορούσε να μάθει σωστά αλλά μπορούσε εντελώς τυχαία να υπολογίσει τις τιμές στα δεδομένα ελέγχου. Ένας πιθανός λόγος για αυτή τη συμπεριφορά είναι το μέγεθος του συνόλου των δεδομένων ελέγχου. Για να επαληθεύσουμε αυτή την υποψία μας αλλά και να ερευνήσουμε αν μπορούμε με κάποια άλλη εκτέλεση να βελτιώσουμε τα αποτελέσματα μας, κάναμε κάποιες δοκιμαστικές εκτελέσεις κρατώντας όλες τις παραμέτρους σταθερές εκτός από την παράμετρο καθορισμού του μεγέθους του συνόλου εκμάθησης. Πιο συγκεκριμένα κάναμε τις ακόλουθες εκτελέσεις:

- 80% δεδομένα εκμάθησης & 20% δεδομένα ελέγχου
- 70% δεδομένα εκμάθησης & 30% δεδομένα ελέγχου
- 60% δεδομένα εκμάθησης & 40% δεδομένα ελέγχου
- 50% δεδομένα εκμάθησης & 50% δεδομένα ελέγχου

Βλέποντας τα παραγόμενα αποτελέσματα συμπεράναμε ότι η αρχική μας υποψία ήταν σωστή και ότι το μέγεθος των δεδομένων ελέγχου επιδρά στην ποιότητα των αποτελεσμάτων μας. Από τις εκτελέσεις αυτές βρήκαμε ότι έχουμε καλύτερα αποτελέσματα όταν το μέγεθος των δεδομένων ελέγχου βρίσκεται κοντά στο 40% και με επομένως στις εκτελέσεις μας κρατήσαμε αυτά τα μεγέθη. Με βάση αυτό κάναμε μια δεύτερη ομάδα εκτελέσεων όπου αυτή την φορά είχαμε αλλαγή στο μέγεθος του αρχείου εκμάθησης αλλά και στον αριθμό γενιών. Αυτό έγινε προκειμένου να εξοικονομήσουμε κάποιο χρόνο αφού από τα αποτελέσματα των πρώτων εκτελέσεων με 500 γενιές είδαμε ότι είχαμε εκμάθηση του δικτύου περίπου στις 300 γενιές. Οι νέες εκτελέσεις θα έχουν 350 γενιές και με βάση την προηγούμενη διαπίστωση δεν περιμένουμε να επηρεαστούν από αυτό τον παράγοντα τα αποτελέσματα μας.

Και για τις δύο ομάδες εκτελέσεων, πρώτα δουλέψαμε με το αρχείο δεδομένων Cosomo και μετά με το αρχείο δεδομένων Desharnais όπου έχουν γίνει οι πιο κάτω εκτελέσεις:

- Βάθος Παραγόμενου Δένδρου: 3, 4, 5, 6 και 7
- Αριθμός Κόμβων Παραγόμενου Δένδρου: 16, 20 και 28

Προκειμένου να κάνουμε αποτίμηση αποτελεσμάτων και να επιλέξουμε από το μεγάλο αριθμό παραγόμενων εξισώσεων αυτές που θεωρούμε ότι είναι οι καλύτερες έχουμε θέσει κάποια κριτήρια για να γίνει η επιλογή αυτή. Και τα δύο κριτήρια έχουν να κάνουν με τις τιμές των συντελεστών λάθους που αποθηκεύονται στο αρχείο εξόδου. Το πρώτο κριτήριο που έχουμε θέσει είναι να έχουμε τιμές συντελεστών λάθους στα δεδομένα εκμάθησης καλύτερους, δηλαδή καλύτερες τιμές, από τις αντίστοιχες τιμές λαθών των δεδομένων ελέγχου. Αυτό το έχουμε θέσει σκεπτόμενοι ότι θεωρητικά ένα σύστημα αποδίδει καλύτερα σε δεδομένα που τα γνωρίζει και λιγότερο καλύτερα σε δεδομένα που βλέπει πρώτη φορά. Αυτό το κριτήριο παρέμεινε σταθερό για όλες τις αριθμητικές εκτελέσεις. Το δεύτερο κριτήριο αφορά κάποια όρια τιμών για κάποιους συντελεστές λάθους. Θα επικεντρωθούμε σε τρεις από αυτούς τους συντελεστές και θέτουμε για τις εκτελέσεις του συνόλου δεδομένων Cosomo να έχουμε:

- Μέσο σφάλμα συσχέτισης < 0.6

- Συντελεστής συσχέτισης > 0.85
- Κανονικοποιημένη ρίζα τετραγωνικού σφάλματος < 0.5

Βλέποντας τα αποτελέσματα του δεύτερου συνόλου, του συνόλου Desharnais πρέπει να αλλάξουμε τις πιο πάνω τιμές και να είμαστε λίγο πιο ελαστικοί αφού οι τιμές που παίρνουμε με αυτό το αρχείο είναι λίγο μεγαλύτερες από αυτές του αρχείου Cosmo. Έτσι για το δεύτερο αρχείο μας θέτουμε τις τιμές:

- Μέσο σφάλμα συσχέτισης < 0.7
- Συντελεστής συσχέτισης > 0.7

5.2.2 Πειραματικές Εκτελέσεις Λογικών Συναρτήσεων

Στις λογικές συναρτήσεις η διαδικασία που ακολουθήθηκε για τους τελεστές και τους τελεστέους είναι λίγο διαφορετική από τις αριθμητικές συναρτήσεις, με το αρχείο δεδομένων και σε αυτή την περίπτωση να αποτελείται από αριθμητικά δεδομένα μόνο. Για να δημιουργήσουμε το σύνολο των τελεστών δουλέψαμε με την ακόλουθη διεργασία: Για κάθε χαρακτηριστικό που υπάρχει στο αρχείο δεδομένων μας βρίσκουμε την ελάχιστη και την μέγιστη τιμή. Αυτές οι δύο τιμές θα μας βοηθήσουν να δημιουργήσουμε για κάθε χαρακτηριστικό δύο τελεστέους χρησιμοποιώντας τους τελεστές του μεγαλύτερου και του μικρότερου ή ίσου. Για παράδειγμα έχουμε το χαρακτηριστικό LOC το οποίο έχει ελάχιστη τιμή το 198 και μέγιστη τιμή το 1150. Οι δύο τελεστέοι που θα δημιουργηθούν είναι ($LOC > 198$) και ($LOC \leq 1150$). Επομένως για κάθε χαρακτηριστικό του αρχείου δεδομένων μας θα έχουμε δύο τελεστέους και επομένως ο αριθμός των τελεστών που αντιστοιχεί σε κάθε αρχείο δεδομένων είναι:

$$Term = 2 * k$$

όπου k είναι ο αριθμός των μεταβλητών του αρχείου δεδομένων.

Μετά από αυτό και βλέποντας τα αποτελέσματα που παίρναμε κάναμε μια αλλαγή στον τρόπο δημιουργίας των τελεστών. Για κάθε μια μεταβλητή του αρχείου δεδομένων θα πάρουμε δύο τιμές στο πλαίσιο [ελάχιστη τιμή ... μέγιστη τιμή] των τιμών της μεταβλητής

και στη συνέχεια με τυχαίο τρόπο θα επιλέξουμε δύο τελεστές. Ο ένας τελεστής θα είναι επιλογή μεταξύ των τελεστών '<' ή '<=' ενώ ο δεύτερος θα είναι επιλογή μεταξύ των τελεστών '>' ή '>='. Με αυτό τον τρόπο δεν χρησιμοποιούμε μόνο 2 τελεστές αλλά επιλογή 2 από 4 διαθέσιμους τελεστές. Επίσης μικραίνουμε το διάστημα των τιμών έτσι ώστε να μην έχουμε μεγάλο αριθμό έργων να ικανοποιούν τον κανόνα.

Όσον αφορά το σύνολο των διαθέσιμων τελεστών έχουμε τις εξής λογικές πράξεις:

- AND (&&)
- OR (||)
- NOT (~)
- NOR
- NAND
- XOR
- IF .. THEN .. ELSE..

Για τις λογικές εκτελέσεις η αρχική σκέψη ήταν να δουλέψουμε με τρόπο έτσι ώστε η συνάρτηση καταλληλότητας να βασίζεται με κάποιο τρόπο στην πραγματική τιμή της προσπάθειας. Η πρώτη υλοποίηση θεωρούσε ότι η συνάρτηση καταλληλότητας κάθε παραγόμενης λύσης θα ήταν η μέση τιμή της πραγματικής προσπάθειας όσων γραμμών εισόδου ικανοποιούσαν την λύση. Επομένως με αυτή την υλοποίηση θέλαμε να παίρνουμε κάθε φορά τη λύση που έχει τη μεγαλύτερη τιμή καταλληλότητας. Η υλοποίηση αυτή όμως δεν είχε τα αναμενόμενα αποτελέσματα αφού οι παραγόμενες λύσεις μπορεί να είχαν σχετικά μεγάλη τιμή καταλληλότητας, όμως δεν ήταν λύσεις που ικανοποιούσαν τις περισσότερες γραμμές εισόδου. Αυτό έγινε αντιληπτό όταν μετά από διάφορες εκτελέσεις παρατηρώντας τα αποτελέσματα είδαμε ότι όλες οι παραγόμενες λύσεις είχαν πολύ χαμηλή αναλογία ευστοχίας αλλά και αναλογία επιτυχίας τόσο στα δεδομένα ελέγχου όσο και στα δεδομένα εκμάθησης. Για να ξεπεράσουμε το εμπόδιο αυτό αλλάξαμε εντελώς την συνάρτηση καταλληλότητας με τέτοιο τρόπο έτσι ώστε να μην βασίζεται αποκλειστικά και μόνο στην πραγματική τιμή της προσπάθειας. Με την καινούργια λογική η συνάρτηση καταλληλότητας θα βασίζεται στην πραγματική τιμή της

προσπάθειας, στον αριθμό των γραμμών που ικανοποιούν την εξίσωση και στο μέγεθος της εξίσωσης (βλέπε 4.3.2.1). Και με αυτή την συνάρτηση καταλληλότητας θέλουμε να πάρουμε τις λύσεις οι οποίες να έχουν όσο το δυνατό μεγαλύτερη τιμή στη συνάρτηση καταλληλότητας τους. Για τις λογικές εκτελέσεις έχουμε χρησιμοποιήσει τα δύο αρχεία που έχουμε χρησιμοποιήσει και στις αριθμητικές, δηλαδή το αρχείο του Cocomo και το αρχείο του Desharnais.

5.2.3 Πειραματικές Εκτελέσεις Κατηγορικών Συναρτήσεων

Οι κατηγορικές συναρτήσεις έχουν πολλά κοινά με τις λογικές συναρτήσεις αφού δουλεύουν με τον ίδιο ακριβώς τρόπο. Το σύνολο των διαθέσιμων τελεστών είναι το ίδιο, όπως επίσης και η συνάρτηση καταλληλότητας. Η μεγάλη διαφορά μεταξύ των λογικών και των κατηγορικών συναρτήσεων έγκειται στον τρόπο με τον οποίο δημιουργούμε τους τελεστές μας. Και σε αυτή την περίπτωση για να δημιουργήσουμε το σύνολο των τελεστών μας δουλέψαμε με κάθε χαρακτηριστικό του αρχείου δεδομένων. Η διαφορά εδώ είναι ότι δεν έχουμε αριθμητικές τιμές επομένως οι λογικές εκφράσεις του μεγαλύτερου ή του μικρότερου δεν ισχύουν. Σε αυτή την περίπτωση κάναμε χρήση της έννοιας της ισότητας δύο μη αριθμητικών χαρακτηριστικών. Πιο συγκεκριμένα για κάθε μεταβλητή από το αρχείο δεδομένων μας, παίρνουμε όλες τις διακριτές τιμές και για κάθε διακριτή τιμή δημιουργούμε ένα τελεστέο. Για παράδειγμα αν η μεταβλητή LANG μπορεί να πάρει δύο τιμές 0 και 1 τότε θα δημιουργηθούν δύο τελεστέοι: (LANG = '0') και (LANG='1'). Συνεπώς σε αυτή την περίπτωση ο αριθμός των τελεστών μπορεί να είναι πολύ μεγάλος ανάλογα με τον αριθμό των μεταβλητών του αρχείου αλλά και με τον αριθμό των διακριτών τιμών κάθε μεταβλητής. Αυτός είναι και ένας λόγος που οι εκτελέσεις αυτού του τύπου είναι χρονοβόρες.

Για τις κατηγορικές συναρτήσεις δουλέψαμε με το αρχείο του ISBSG, όμως λόγω του ότι αυτό το αρχείο είναι πολύ μεγάλο τόσο σε μεταβλητές όσο και σε έργα η όλη διαδικασία είναι αρκετά χρονοβόρα και δεν μπορέσαμε να έχουμε τόσες πολλές εκτελέσεις όπως με τις υπόλοιπες ομάδες.

5.3 Παρουσίαση Αποτελεσμάτων

Μέσα από τα διάφορα πειράματα που έχουμε αναφέρει πιο πάνω πήραμε για κάθε εκτέλεση διάφορα αποτελέσματα. Παρατηρήσαμε ότι παρόλο που οι παράμετροι μας ήταν οι ίδιοι για τα διάφορα αρχεία δεδομένων, εντούτοις τα αποτελέσματα μεταξύ ίδιων εκτελέσεων είναι αρκετά διαφορετικά. Ας δούμε και να αναλύσουμε όμως τα αποτελέσματα που έχουμε πάρει στις διάφορες εκτελέσεις μας.

5.3.1 Αποτελέσματα Αριθμητικών Εκτελέσεων

Όπως έχουμε ήδη αναφέρει στις αριθμητικές εκτελέσεις έχουμε κάνει συνολικά οχτώ εκτελέσεις, 5 εκτελέσεις με βάση το βάθος του παραγόμενου δένδρου και 3 εκτελέσεις με βάση τον αριθμό των κόμβων του παραγόμενου δένδρου. Η κάθε μια εκτέλεση αποτελείται από 100 επαναλήψεις (shuffles) και επομένως για κάθε εκτέλεση έχουμε ένα αρχείο εξόδου με 100 εξισώσεις – την καλύτερη εξίσωση για κάθε μια επανάληψη.

Για να μπορέσουμε να επιλέξουμε τα καλύτερα αποτελέσματα από το σύνολο όλων αυτών των εκτελέσεων σαν πρώτο κριτήριο θεωρούμε το γεγονός ότι τα δεδομένα εκμάθησης πρέπει να έχουν καλύτερες τιμές από τα δεδομένα ελέγχου. Δεύτερος έλεγχος είναι να έχουμε τιμές στους συντελεστές λάθους οι οποίοι να είναι αποδεχτοί από εμάς ως καλές τιμές:

- Μέσο σφάλμα συσχέτισης < 0.6
- Συντελεστής συσχέτισης > 0.85
- Κανονικοποιημένη ρίζα τετραγωνικού σφάλματος < 0.5

Ας πάρουμε την ομάδα εκτελέσεων όπου το μέγεθος του συνόλου των δεδομένων ελέγχου είναι το 20% του αρχείου εισόδου (Cocomo A). Οι εκτελέσεις με βάθος δένδρου 3 και βάθος 6 δεν μας έδωσαν κανένα καλό αποτέλεσμα που να είναι στα πλαίσια των τιμών που θέσαμε ως κριτήρια και αυτός είναι και ο λόγος που δεν αναφέρονται στους πίνακες αποτελεσμάτων. Τα αποτελέσματα που έχουμε πάρει για το αρχείο δεδομένων του Cocomo σε αυτήν την πρώτη ομάδα εκτελέσεων υπάρχουν στο παράρτημα Α στο

τέλος της μελέτης αυτής. Προκειμένου να μπορέσουμε να εξηγήσουμε και να αναλύσουμε τον τρόπο που κάναμε αποτίμηση δεδομένων θα κάνουμε χρήση μιας εκτέλεσης μόνο. Στην προκειμένη περίπτωση ας πάρουμε για παράδειγμα την εκτέλεση όπου είχαμε επιλογή τα παραγόμενα δένδρα να έχουν μέγιστο βάθος 5 και συγκεκριμένα την εκτέλεση αριθμός 11.

	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Δεδομένα Εκμάθησης	0.138218	0.991288	63670.176505	0.484581	125.524421	0.313725	0.352811
Δεδομένα Ελέγχου	0.242901	0.978594	205973.586974	0.493973	249.402638	0.25	1.199231

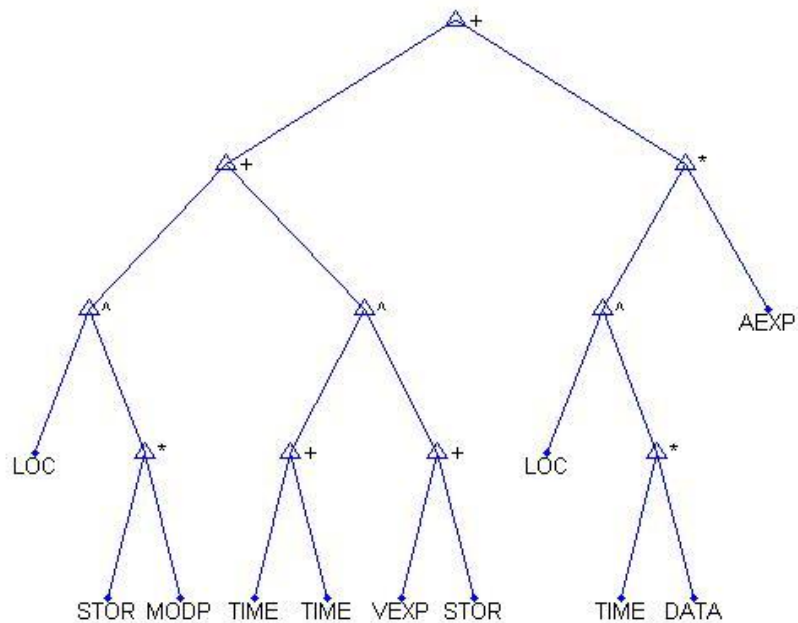
Πίνακας 5.1: Αποτελέσματα Εκτέλεση 11 – Πειραματικής ομάδας Α

Στα πιο πάνω αποτελέσματα βλέπουμε ότι οι τιμές των συντελεστών λάθους για τα δεδομένα εκμάθησης είναι καλύτερες από τις αντίστοιχες τιμές των δεδομένων ελέγχου. Αυτό ικανοποιεί την πρώτη προϋπόθεση που έχουμε θέσει για αποτίμηση αποτελεσμάτων και επιλογή των καλύτερων εκτελέσεων. Για τον συντελεστή NRMSE έχουμε μια αρκετά χαμηλή τιμή $NRMSE=0.24$ που είναι πολύ κοντά στο 0, την ιδανική τιμή που θέλουμε να παίρνουμε. Παράλληλα και ο συντελεστής $CC=0.98$ έχει πολύ καλή τιμή αφού πλησιάζει αρκετά στο 1, την ιδανική τιμή που θέλουμε να έχει αυτός ο συντελεστής. Ο τρίτος συντελεστής το MRE έχει τιμή κάτω από το όριο που θέσαμε στις προϋποθέσεις μας, δηλαδή $MRE=0.49$.

Η αριθμητική εξίσωση που αντιστοιχεί στο πιο πάνω αποτέλεσμα είναι η ακόλουθη:

$$(((LOC^{(STOR*MODP)}))+((TIME+TIME)^{(VEXP+STOR)}))+((LOC^{(TIME*DATA)})*AEXP))$$

Για να είναι όμως και οπτικά αισθητή η μορφή αυτής της εξίσωσης που έχει παρθεί κατά την πειραματική μας διαδικασία, η εικόνα 5.2 που ακολουθεί δείχνει σε μορφή δυαδικού δένδρου πως αναπαριστάται η εξίσωση που έχουμε αναφέρει.



Εικόνα 5.2: Αναπαράσταση σε δένδρο

Συγκρίνοντας τις δύο πειραματικές εκτελέσεις που έχουμε κάνει με το αρχείο δεδομένων Cocomo, παρατηρούμε ότι στην εκτέλεση όπου αυξήσαμε το σύνολο των δεδομένων ελέγχου έχουμε καταφέρει να αυξήσουμε τον αριθμό των αποτελεσμάτων τα οποία ικανοποιούν το πρώτο κριτήριο που έχουμε θέσει, δηλαδή οι συντελεστές λάθους των δεδομένων εκμάθησης να είναι καλύτεροι από αυτούς των δεδομένων ελέγχου. Αναφορικά με τις τιμές των συντελεστών δεν υπήρξε καμία διαφορά και επομένως το μέγεθος των δεδομένων ελέγχου δεν επηρεάζει τα αποτελέσματα μας.

Ανάλογη συμπεριφορά είχαμε και με την επόμενη ομάδα εκτελέσεων με χρήση του αρχείου Desharnais. Και εδώ είχαμε τις δύο πειραματικές ομάδες που αναφέραμε πιο πάνω και επίσης τα συμπεράσματά μας είναι τα ίδια αναφορικά με τη σύγκριση των αποτελεσμάτων των δύο ομάδων εκτελέσεων. Πίνακες με τα καλύτερα αποτελέσματα των εκτελέσεων με αυτό το αρχείο υπάρχουν στο παράρτημα Α στο τέλος της παρούσας μελέτης.

Στις αριθμητικές εκτελέσεις που κάναμε και με τα δύο αρχεία δεδομένων, παρατηρούμε ότι έχουμε αρκετά καλύτερα αποτελέσματα με το αρχείο του Cocomo παρά με το Desharnais. Γενική επίσης παρατήρηση είναι και το γεγονός ότι ανεξάρτητα με το αρχείο εισόδου τα περισσότερα αποτελέσματα που πήραμε είχαν τιμές συντελεστών λάθους για τα δεδομένα ελέγχου καλύτερες από τις αντίστοιχες των δεδομένων εκμάθησης, γεγονός που αξίζει περισσότερης διερεύνησης αφού είναι ένα γεγονός το οποίο ίσως να δηλώνει τυχαία καλή συμπεριφορά του συστήματος. Πιθανές αιτίες μπορεί να είναι το μέγεθος των δεδομένων ελέγχου και εκμάθησης, κάτι που μέσα από την διαδικασία της δεύτερης ομάδας εκτελέσεων που έχουμε κάνει φαίνεται να έχει κάποια βάση αφού το φαινόμενο αυτό με τις τιμές των συντελεστών στα δεδομένα εκμάθησης και ελέγχου είχε μειωθεί με την αύξηση του ποσοστού των δεδομένων ελέγχου. Άλλος πιθανός παράγοντας ίσως να είναι και το γεγονός ότι σε κάθε εκτέλεση έχουμε εντελώς τυχαίο διαχωρισμό των δεδομένων μας ή ίσως ακόμη να επηρεάζει και το συνολικό μέγεθος (σε έργα) του αρχείου εισόδου.

5.3.2 Αποτελέσματα Λογικών Εκτελέσεων

Αναφορικά με τις λογικές εκτελέσεις όπως έχουμε πει η συνάρτηση καταλληλότητας βασίζεται όχι μόνο στην τιμή της πραγματικής προσπάθειας, αλλά αποτελεί συνδυασμό διαφόρων παραγόντων, ένας εκ των οποίων είναι και το μέγεθος του παραγόμενου δένδρου (βλέπε 4.3.2.1). Με βάση αυτό θα δείξουμε τα αποτελέσματα των λογικών συναρτήσεων.

Οι εκτελέσεις των λογικών εξισώσεων έγιναν στα ίδια αρχεία που αναφέραμε πιο πάνω και υπάρχουν στο παράρτημα Α στο τέλος της αναφοράς.

Βλέποντας κανείς τα αποτελέσματα, σίγουρα θα διερωτηθεί πως είναι δυνατόν να έχουμε τόσα πολλά αποτελέσματα με ποσοστά ευστοχίας αλλά και επιτυχίας στο 100%. Παρόλο που έχουμε αλλάξει την συνάρτηση καταλληλότητας μας (βλέπε 4.3.2.1) να λαμβάνει υπόψη διάφορους παράγοντες εντούτοις όμως υπάρχει εξήγηση του φαινομένου αυτού. Η εξήγηση βασίζεται στο γεγονός ότι οι τελεστέοι που

δημιουργούνται σε αυτές τις περιπτώσεις είναι γενικοί. Όπως έχουμε εξηγήσει και στο κεφάλαιο 4, για να δημιουργήσουμε τους τελεστές των λογικών εκτελέσεων παίρνουμε τις ελάχιστες και μέγιστες τιμές κάθε μεταβλητής και δημιουργούμε δύο τελεστές για κάθε μια. Κατά συνέπεια οι κανόνες που θα δημιουργηθούν κατά τη λειτουργία του γενετικού αλγορίθμου έχουν μεγάλο εύρος και άρα συμπεριλαμβάνουν - ικανοποιούν πολλά έργα. Ας πάρουμε για παράδειγμα την μεταβλητή LOC. Έστω ότι αυτή η μεταβλητή έχει ελάχιστη τιμή 1.98 και μέγιστη 1150. Βλέπουμε ότι το εύρος των τιμών είναι πολύ μεγάλο και επομένως κανόνες που συμπεριλαμβάνουν αυτή τη μεταβλητή θα έχουν ψηλά ποσοστά ευστοχίας. Η λειτουργία αυτή των λογικών εξισώσεων μπορεί να θεωρηθεί και μελλοντική εργασία για να περιοριστεί κάπως το εύρος των τιμών αυτών.

5.3.3 Αποτελέσματα Κατηγορικών Εκτελέσεων

Οι εκτελέσεις για αυτήν την ομάδα έχουν γίνει με τα αρχείο δεδομένων ISBSG. Στο αρχείο αυτό υπάρχει κάποιες μεταβλητές οι οποίες είναι αριθμητικές. Αυτές τις στήλες τις έχουμε αφαιρέσει και χρησιμοποιήσαμε τις υπόλοιπες για τις κατηγορικές εκτελέσεις.

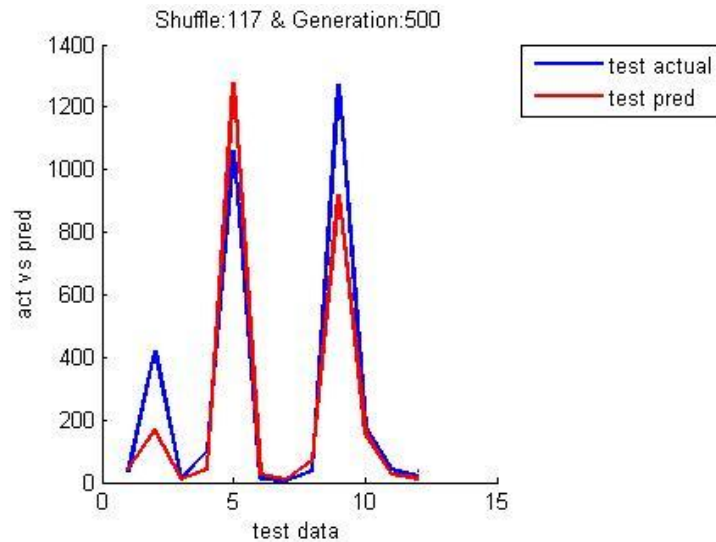
Στα αποτελέσματα που πήραμε (παράρτημα Α) και εδώ οι περισσότερες εξισώσεις που έχουμε πάρει έχουν ψηλές τιμές αναλογιών ευστοχίας και επιτυχίας. Σε αυτές τις εκτελέσεις δεν έχουμε το φαινόμενο που είχαμε και στις λογικές με τις εξισώσεις να είναι σε γενικευμένη μορφή, αφού εδώ δεν έχουμε διαστήματα αριθμητικών τιμών. Οι εκτελέσεις που κάναμε είχαν 20% του συνόλου των δεδομένων σαν δεδομένα έλεγχο και το υπόλοιπο 80% για δεδομένα εκμάθησης. Με αυτό το τρόπο για το αρχείο δεδομένων του ISBSG είχαμε 374 έργα για εκμάθηση και 93 έργα για έλεγχο.

5.4 Συμπεράσματα και Γραφικές Παραστάσεις

Όπως έχουμε ήδη αναφέρει, τα αποτελέσματα που πήραμε στις διάφορες εκτελέσεις είναι πάρα πολλά, οπότε έγινε μια εξαγωγή και παρουσιάστηκαν σε πίνακες, ένα υποσύνολο αυτών των αποτελεσμάτων που θεωρούνται ότι είναι τα καλύτερα δυνατά αποτελέσματα.

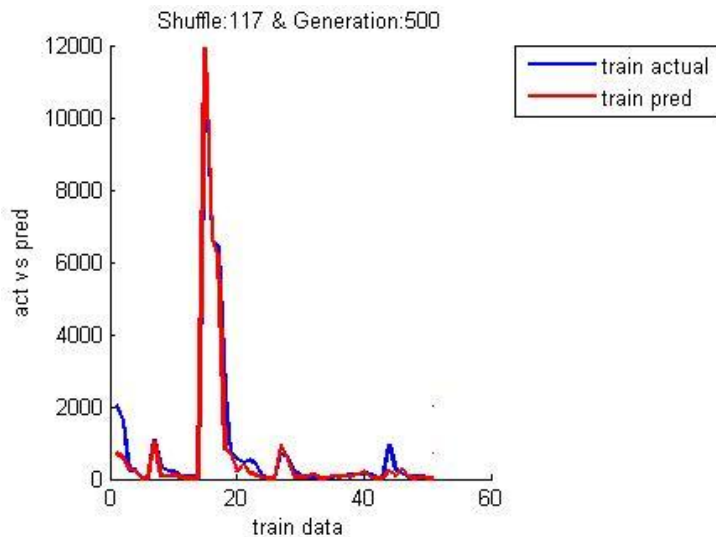
Βασική αρχή που ακολουθήθηκε ήταν να έχουμε τις τιμές για τους συντελεστές λάθους MRE, CC και NRMSE για τα δεδομένα εκμάθησης πιο καλές από τις αντίστοιχες των δεδομένων ελέγχου. Καλές τιμές εννοούμε να έχουμε τιμές MRE και NRMSE στα δεδομένα εκμάθησης μικρότερες από αυτές των δεδομένων ελέγχου και τιμή CC μεγαλύτερη στα δεδομένα εκμάθησης από τα δεδομένα ελέγχου. Βλέποντας τις εξισώσεις που πήραμε παρατηρήσαμε ότι μεγάλο ποσοστό τους δεν ικανοποιούσε την βασική αυτή αρχή. Αυτό σημαίνει ότι στις περιπτώσεις αυτές όταν είχαμε τιμές σε αυτούς τους συντελεστές καλές (<0.5 για MRE και NRMSE ή >0.85 για CC) αυτό σημαίνει ότι το σύστημα δεν είχε εκπαιδευτεί, όμως μπορούσε να υπολογίσει σωστά στα δεδομένα ελέγχου τυχαία. Παράγοντας που επηρεάζει το γεγονός αυτό αποτελεί το μέγεθος των δεδομένων ελέγχου. Αυξάνοντας το ποσοστό για τα δεδομένα ελέγχου παρατηρήσαμε βελτίωση, μείωση δηλαδή του αριθμού των εξισώσεων που έχουν καλύτερους συντελεστές στα δεδομένα ελέγχου παρά στα δεδομένα εκμάθησης. Πιθανός παράγοντας μπορεί να αποτελεί και το γεγονός ότι στα δεδομένα εισόδου μας είχαμε κάποια έργα με πραγματική τιμή προσπάθειας πολύ ψηλή σε σχέση με τα υπόλοιπα δεδομένα μας (outlier). Αυτό όμως είναι κάτι το οποίο πρέπει να εξεταστεί εξονυχιστικά, αφαιρώντας δηλαδή αυτού του είδους δεδομένα από το σύνολο μας και μέσω του συστήματος να γίνουν οι διάφορες εκτελέσεις που κάναμε και στο τέλος να συγκρίνουμε τα αποτελέσματα μεταξύ τους.

Με τη βοήθεια της διεπαφής που έχουμε υλοποιήσει μπορούμε να δούμε κάποιες γραφικές παραστάσεις των τελικών εξισώσεων που μας έδωσαν οι εκτελέσεις μας. Πιο κάτω θα δούμε και θα σχολιάσουμε τις γραφικές παραστάσεις για μια εκτέλεση με το αρχείο Cocomo η οποία έδωσε πολύ καλές τιμές στους συντελεστές λάθους. Η εκτέλεση αυτή αναφέρεται στον πίνακα 5.1 και είναι η εκτέλεση 117 από την ομάδα εκτελέσεων με βάθος παραγόμενου δένδρου 4.



Γραφική Παράσταση 5.1

Σε αυτή την γραφική παράσταση μπορούμε να δούμε σύγκριση των τιμών για τα δεδομένα ελέγχου μεταξύ της πραγματικής τιμής προσπάθειας και της υπολογιζόμενης. Σε γενικές γραμμές βλέπουμε ότι η πρόβλεψη είναι επιτυχής αφού είναι λίγα τα σημεία (2ο, 5ο και 9ο) στα οποία δεν έγινε εύστοχος υπολογισμός της προσπάθειας. Πιο κάτω βλέπουμε την ίδια γραφική παράσταση για τα δεδομένα εκμάθησης. Θεωρητικά θα πρέπει να έχουμε πολύ καλύτερη σύγκλιση αποτελεσμάτων σε αυτά τα δεδομένα.

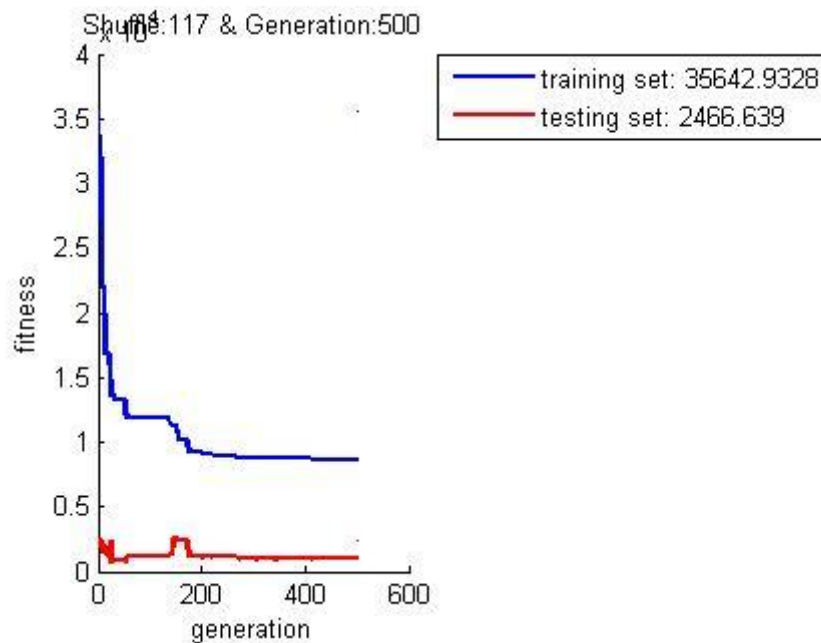


Γραφική Παράσταση 5.2

Όπως αναμέναμε στα δεδομένα εκμάθησης έχουμε περισσότερη σύγκλιση στις πραγματικές και υπολογιζόμενες τιμές. Επίσης εδώ στα σημεία όπου έχουμε κάποια διαφορά στις τιμές βλέπουμε ότι αυτή δεν είναι πολύ μεγάλη και σίγουρα είναι

μικρότερη σε σχέση με την προηγούμενη γραφική παράσταση των δεδομένων ελέγχου. Με αυτές τις δύο γραφικές παραστάσεις μπορούμε να πούμε ότι στη συγκεκριμένη εκτέλεση είχαμε επιτυχία στην εκμάθηση του συστήματος.

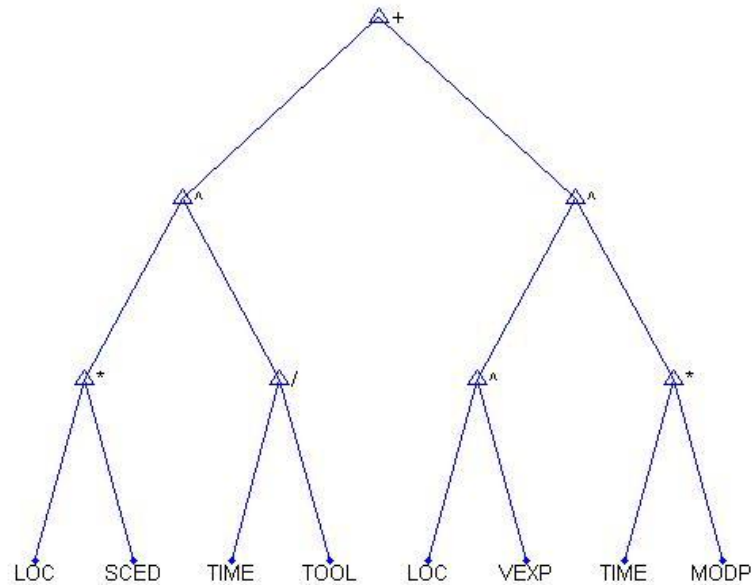
Η επόμενη γραφική παράσταση μας δείχνει την εξέλιξη της τιμής καταλληλότητας σε κάθε γενιά στο συγκεκριμένο παράδειγμα εκτέλεσης που αναφερόμαστε.



Γραφική Παράσταση 5.3

Αυτή η παράσταση μας δείχνει ότι αναφορικά με τα δεδομένα εκμάθησης αρχικά η τιμή της καταλληλότητας είναι πολύ ψηλή η οποία όμως από την πρώτη γενιά μειώνεται αισθητά. Αυτό συμβαίνει γιατί όπως έχουμε ήδη αναφέρει, το προτεινόμενο σύστημα κάνει αρχικοποίηση του πληθυσμού με εντελώς τυχαίο τρόπο. Αυτό είναι ένα φαινόμενο που θα το παρατηρήσει κανείς σε όποια εκτέλεση και αν επιλέξει να δει την γραφική παράσταση της τιμής καταλληλότητας. Σε κάθε γενιά με την δημιουργία νέων ατόμων η τιμή καταλληλότητας μειώνεται συνεχώς μέχρι περίπου τη γενιά 300 όπου αρχίζει να έχει μια σταθερή τιμή μέχρι το τέλος της εκτέλεσης.

Για την συγκεκριμένη εκτέλεση που βλέπουμε, ας δούμε πως αναπαριστάται σε μορφή δένδρου και το άτομο με την καλύτερη τιμή καταλληλότητας.



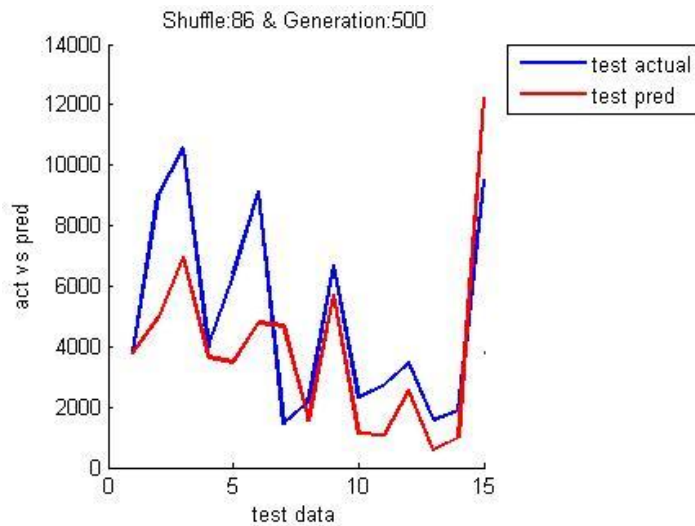
Εικόνα 5.2

Η αριθμητική εξίσωση που αντιστοιχεί στην πιο πάνω γραφική απεικόνιση είναι η ακόλουθη:

$$(((LOC*SCED)^(TIME/TOOL)) + ((LOC^VEXP)^(TIME*MODP)))$$

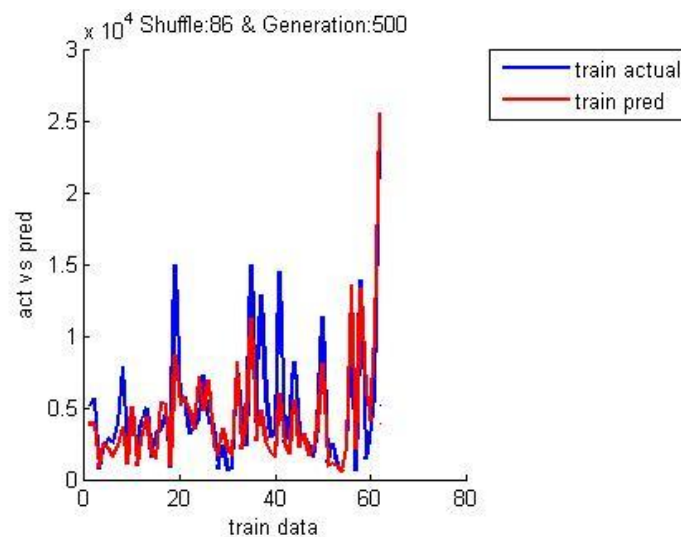
Μπορούμε εύκολα να διακρίνουμε ότι το βάθος του δένδρου αυτού είναι 4. Επίσης βλέπουμε ότι στα φύλλα του δένδρου έχουμε κάποιο από τους τελεστές μας (χαρακτηριστικό του αρχείου εισόδου), ενώ στους ενδιάμεσους κόμβους του δένδρου έχουμε τους αριθμητικούς τελεστές που λαμβάνουν μέρος σε αυτή την εξίσωση.

Όπως έχουμε δει στα αποτελέσματα μας πιο πάνω με το αρχείο δεδομένων Desharnais δεν είχαμε τόσο καλά αποτελέσματα όσο με το Cocomo. Το σύστημα μας δηλαδή δεν κατόρθωσε να έχει τα ποσοστά εκμάθησης που είχαμε στην προηγούμενη περίπτωση. Αυτό θα πρέπει να είναι εμφανές και στις αντίστοιχες γραφικές παραστάσεις. Για να δούμε αν όντως αυτό ισχύει επιλέξαμε να δούμε παραστάσεις για την εκτέλεση 86 της ομάδας εκτελέσεων με βάθος 5.



Γραφική Παράσταση 5.4

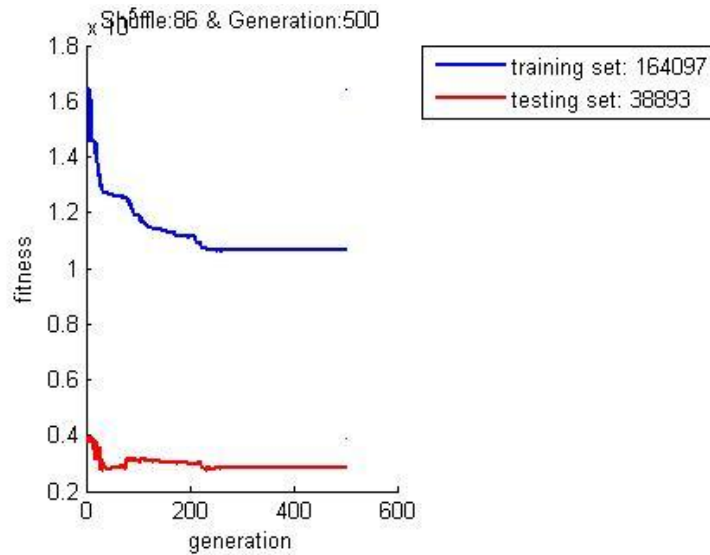
Αυτό που αναφέραμε φαίνεται καθαρά και στην πιο πάνω γραφική παράσταση της πραγματικής και της υπολογιζόμενης τιμής καταλληλότητας. Σε αυτή διακρίνουμε ότι τα σημεία επιτυχούς υπολογισμού της τιμής καταλληλότητας είναι λίγα. Κάτι ανάλογο θα πρέπει να συμβαίνει και με τα δεδομένα εκμάθησης σε πιο μικρή όμως εμβέλεια.



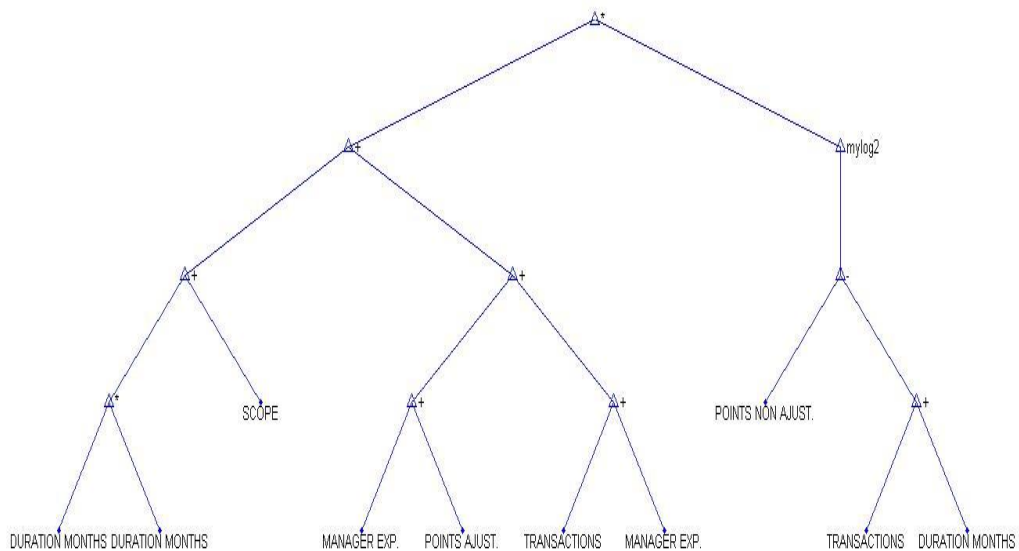
Γραφική Παράσταση 5.5

Και στα δεδομένα εκμάθησης έχουμε αποκλίσεις τιμών από τις πραγματικές η διαφορά όμως των τιμών είναι μικρότερης κλίμακας. Στην γραφική παράσταση της καταλληλότητας βλέπουμε ότι η τιμή καταλληλότητας μειώνεται όσο περνούν οι γενιές και περίπου στην γενιά 220 έχουμε σταθεροποίηση της τιμής καταλληλότητας. Παρόλα

αυτά όμως η τιμή της καταλληλότητας παραμένει σε ψηλές τιμές και δεν μειώνεται κάτω από την τιμή 1.



Γραφική Παράσταση 5.6



Εικόνα 5.3

Η εικόνα 5.3 είναι η αναπαράσταση της εξίσωσης της εκτέλεσης 86 του καλύτερου ατόμου σε μορφή δένδρου. Η παραγόμενη εξίσωση είναι:

$$(((\text{DURATION MONTHS} * \text{DURATION MONTHS}) + \text{SCOPE}) + ((\text{MANAGER EXP.} + \text{POINTS AJUST.}) + (\text{TRANSACTIONS} + \text{MANAGER EXP.}))) * ((\text{POINTS NON AJUST.} - (\text{TRANSACTIONS} + \text{DURATION MONTHS})))$$

		Cocomo A		Cocomo B		Desharnais A		Desharnais B	
		Training	Testing	Training	Testing	Training	Testing	Training	Testing
NRMSE	Min	0.086510	0.165555	0.062762	0.242058	0.562001	0.693829	0.501947	0.633903
	Max	0.378665	0.597460	0.527362	0.595039	0.682441	0.939018	0.684264	0.890040
	Average	0.190090	0.431344	0.148131	0.488666	0.632146	0.759460	0.621503	0.714452
CC	Min	0.928760	0.855426	0.867468	0.843162	0.733768	0.701294	0.740410	0.700485
	Max	0.996509	0.988188	0.998239	0.986934	0.836068	0.773332	0.868362	0.792431
	Average	0.981551	0.942763	0.987057	0.931879	0.777960	0.732439	0.787212	0.731292
MSE	Min	24949.44408	5343.652725	7119.083351	22323.4739	5626059.903	1933337.629	3966909.278	4606103.704
	Max	577935.8158	3165995.991	158544.9377	2603386.6	9442534.217	13718799.71	10217903.06	15005922.53
	Average	123624.0232	591757.8109	69106.7751	729339.662	7275020.333	8443709.033	6594097.288	9512125.565
MRE	Min	0.418647	0.468547	0.352763	0.470037	0.455544	0.485280	0.412020	0.520538
	Max	0.553568	0.599758	0.564873	0.597925	0.644675	0.689460	0.637515	0.686154
	Average	0.478008	0.539434	0.463950	0.533933	0.547644	0.618068	0.530778	0.623261
MAE	Min	79.517178	49.562412	52.597095	91.351973	1586.149723	1108.060982	1379.928089	1594.1540053
	Max	242.188434	985.964641	179.757610	658.989495	1914.128129	2651.611708	2193.243688	2592.407486
	Average	150.608095	315.443648	124.182426	332.300413	1780.428142	2096.349322	1702.679732	2174.788238
PRED	Min	0.156862	0.0833333	0.210526	0.08	0.322580	0.066666	0.255319	0.066666
	Max	0.450980	0.5	0.5	0.36	0.483870	0.466666	0.489361	0.5
	Average	0.322303	0.223958	0.328947	0.235555	0.411910	0.276923	0.406619	0.307407
BRE	Min	0.046378	0.281301	0.042043	0.019096	0.081957	0.008909	0.005042	0.095417
	Max	2.160979	2.203822	3.292817	2.777046	0.360195	0.452886	0.364266	0.341629
	Average	0.684914	0.907760	0.637591	0.802930	0.130992	0.141360	0.124856	0.167068

Πίνακας 5.28: Ελάχιστες - Μέγιστες - Μέσες Τιμές

Στον πιο πάνω πίνακα βλέπουμε συνοπτικά τις ελάχιστες, μέγιστες αλλά και μέσες τιμές για κάθε ένα από τους συντελεστές λάθους σε κάθε μια ομάδα εκτελέσεων που έχουμε κάνει. Ο μέσος όρος των τιμών του αρχείου Cocomo είναι σαφέστατα βελτιωμένες από τις αντίστοιχες τιμές του αρχείου Desharnais. Επίσης συγκρίνοντας τις τιμές μεταξύ των ίδιων αρχείων παρατηρούμε ότι με την αύξηση του μεγέθους του συνόλου των δεδομένων ελέγχου έχουμε κάπως βελτιωμένα αποτελέσματα. Αυτό είναι πιο εμφανές στο δεύτερο αρχείο όπου οι διαφορές στις τιμές είναι πιο μεγάλες.

Αναφορικά με το σύνολο δεδομένων του Cocomo, οι Martin Lefley and Martin J. Shepperd [19] έχουν κάνει εκτελέσεις γενετικού αλγορίθμου χρησιμοποιώντας τους ίδιους συντελεστές λάθους με την παρούσα μελέτη. Η έρευνα αυτή εστιάζοταν στην χρήση διάφορων τεχνικών σε διάφορα σύνολα δεδομένων με σκοπό να μοντελοποιήσουν και να υπολογίσουν την προσπάθεια ανάπτυξης λογισμικού. Αναφορικά με τον γενετικό προγραμματισμό η μελέτη εστιάζοταν στο να καθορίσει κατά πόσο ο γενετικός προγραμματισμός δίνει καλύτερες λύσεις όταν λαμβάνει υπόψη δεδομένα που αναφέρονται ειδικά στην εταιρεία ή αν μπορεί να δώσει καλύτερες λύσεις σε γενικά δεδομένα. Ένα από τα συμπεράσματα της μελέτης αυτής είναι ότι ο γενετικός προγραμματισμός αποδίδει σχετικά καλά αλλά είναι πιο δύσκολο να τον διαμορφώσεις και επίσης δίνει πιο περίπλοκες λύσεις. Ας δούμε τα αποτελέσματα τους και να συγκρίνουμε τις δύο εργασίες όσο αφορά τα αποτελέσματα του γενετικού προγραμματισμού.

	Estimated effort						
	Random	LSR	1-NN	3-NN	ANN	GP	Average
correlation	-0.161	0.846	0.390	0.497	0.806	0.937	0.890
AMSE	28.091	4.601	13.970	14.720	6.584	1.981	3.596
Pred(25)%	13.333	40.000	33.333	20.000	26.667	40.000	33.333
MMRE	166.39	46.925	85.401	59.192	68.856	37.670	43.467
BMMRE	238.35	73.629	110.022	79.049	85.478	62.797	47.343
Worst case MRE	609	150	332	185	195	125	167

Εικόνα 5.5: Αποτελέσματα άλλων ερευνητών [19]

Βλέποντας τα πιο πάνω αποτελέσματα παρατηρούμε ότι συγκρίνοντας τα δικά μας αποτελέσματα με αυτά της πιο πάνω έρευνας οι τιμές των συντελεστών του λάθους είναι πολύ κοντινές μεταξύ τους. Στην πιο πάνω έρευνα το MRE = 37.67 σε αντίθεση με το δικό μας που είναι λίγο πιο ψηλό 46.85. Ο συντελεστής CC = 0.937 ενώ η τιμή των δικών μας εκτελέσεων είναι 0.988 που είναι λίγο καλύτερη. Από την άλλη ο συντελεστής PRED έχει πιο ψηλές τιμές στην αναφερόμενη έρευνα PRED = 40 ενώ οι εκτελέσεις μας είναι λίγο καλύτερες αφού έχουμε τιμή PRED=50. Γενικά βλέπουμε ότι τα αποτελέσματα των δύο ερευνών συγκλίνουν.

Στις λογικές εκτελέσεις έχουμε δει στα αποτελέσματα μας ψηλές τιμές αναλογίας ευστοχίας και επιτυχίας. Το φαινόμενο αυτό συμβαίνει γιατί οι λογικές εξισώσεις που δημιουργούνται χρησιμοποιούν τελεστές με μεγάλο εύρος τιμών. Αυτό έχει ως αποτέλεσμα μεγάλος αριθμός έργων να ικανοποιούν πολλές εξισώσεις και ως αποτέλεσμα έχουμε αυτά τα ψηλά ποσοστά.

Περνώντας τώρα στο σύνολο δεδομένων του Desharnais, οι C.J. Burges και M.Lefley [5] έχουν χρησιμοποιήσει το αρχείο δεδομένων Desharnais και με τη βοήθεια γενετικού προγραμματισμού έχουν κάνει πειράματα. Στόχος της μελέτης αυτής ήταν να ερευνήσουν τις δυνατότητες που μπορεί να έχει ο γενετικός προγραμματισμός στον υπολογισμό της προσπάθειας ανάπτυξης λογισμικού. Με αυτή την μελέτη οι ερευνητές έχουν αποδείξει ότι ο γενετικός προγραμματισμός μπορεί να παρέχει βελτιωμένη ακρίβεια εξαρτώμενη από το μέτρο αποτίμησης της ακρίβειας. Πιο κάτω φαίνονται τα αποτελέσματα που έχουν πάρει:

	Artificial neural network			Genetic programming		
	Worst	Average	Best	Worst	Average	Best
Correlation	0.588	0.635	0.650	0.612	0.752	0.824
AMSE	6.278	5.477	5.209	14.58	11.13	7.77
Pred(25)	10	10	10	2	4.2	5
Pred(25)%	56	56	56	11.2	23.5	28
MMRE	65.45	60.63	59.23	52.12	44.55	37.95
BMMRE	74	69	66	92.47	74.57	59.23

Εικόνα 5.4: Αποτελέσματα άλλων ερευνητών [5]

Από τις τιμές που φαίνονται στην πιο πάνω εικόνα μπορούμε να κάνουμε μια σύγκριση των αποτελεσμάτων της δικής μας πειραματικής εργασίας και αυτής της έρευνας. Βλέπουμε ότι σε αυτή την έρευνα για CC έχουμε τιμή μέσο όρο 0.752 σε σύγκριση με την δική μας που είναι 0.731 είναι δύο κοντινές εκτιμήσεις. Κοντινές είναι επίσης και οι τιμές για την καλύτερη τιμή. Για το συντελεστή κόστους MRE η σχετική έρευνα έχει δώσει μέσο όρο 44.55 ενώ εμείς έχουμε 61.8.

Επίσης ο J.J. Dolado [7], μεταξύ άλλων τεχνικών, έχει χρησιμοποιήσει και γενετικό προγραμματισμό σε ορισμένα αρχεία δεδομένων συμπεριλαμβανομένου και του Desharnais. Η έρευνα του ερευνητή εστιάζεται στο να χρησιμοποιήσει γενετικό προγραμματισμό και να εξερευνήσει διάφορες συναρτήσεις καταλληλότητας. Στην εικόνα 5.6 μπορούμε να δούμε τις τιμές για MRE και PRED.

Data sets		Curve estimation	Genetic programming	Marginal cost
<i>Desharnais [12]</i>				
K	Equation	$21.47 \times fp^{0.935265}$	$32.9 \times fp^{0.8854}$	Low economies of scale
	PRED(0.25)	44.26	51.6	
	MMRE	0.5428	0.623	

Εικόνα 5.6: Αποτελέσματα άλλων ερευνητών [7]

Συγκρίνοντας τις τιμές αυτής της έρευνας με την δική μας βλέπουμε ότι τα αποτελέσματα όσο αφορά το συντελεστή MRE είναι πάρα πολύ κοντινά. Σε αυτή την έρευνα έχουμε MRE=0.623 και στα δικά μας αποτελέσματα έχουμε MRE=0.618. Παράλληλα με αυτό το συντελεστή τα αποτελέσματα που έχουμε πάρει για το συντελεστή PRED είναι και αυτά κοντινά. Στην δική μας πειραματική διαδικασία έχουμε PRED=46.6 και στην έρευνα του ο Dolado έχει PRED=51.6.

Οι παραγόμενες εξισώσεις μας δείχνουν ότι είναι γενικές εξισώσεις αφού μπορούν να επαληθεύσουν το μεγαλύτερο μέρος των δεδομένων ελέγχου, αλλά επίσης είναι και εξισώσεις που αντιπροσωπεύουν τα δεδομένα μας. Με αυτές μπορούμε να ελέγξουμε και να τις επαληθεύσουμε και με άλλα αρχεία δεδομένων που έχουν τις ίδιες μεταβλητές.

Εκτός από τα συμπεράσματα αυτά, η πειραματική αυτή μελέτη μας έχει δώσει και κάποιες ενδείξεις οι οποίες χρειάζονται περαιτέρω διερεύνηση προκειμένου να επαληθευτούν ή ακόμα και να διαψευθούν. Πιο συγκεκριμένα οι ενδείξεις αφορούν κάποια χαρακτηριστικά τα οποία ίσως να παίζουν σημαντικό ρόλο στην προσπάθεια υπολογισμού του κόστους ανάπτυξης λογισμικού. Οι ενδείξεις αυτές είναι πάρα πολύ σημαντικές εφόσον αποτελούν σημαντικοί παράγοντες που επηρεάζουν το κόστος ανάπτυξης λογισμικού. Η γνώση αυτών των παραγόντων θα βοηθήσει πολύ τους

διαχειριστές της ανάπτυξης λογισμικού στα πρώτα στάδια της διαδικασίας. Θα μπορούν με αυτό τον τρόπο να κάνουν ένα αρχικό υπολογισμό γνωρίζοντας τις τιμές αυτών των μεταβλητών για το συγκεκριμένο λογισμικό και με αυτό τον τρόπο θα έχουν ένδειξη στο που περίπου θα κυμαίνεται το κόστος για το συγκεκριμένο λογισμικό. Έτσι αν το αρχικό αυτό κόστος ξεπερνά κατά πολύ τις προσδοκίες τότε θα μπορούν να κάνουν αλλαγές ή να στραφούν σε άλλες κατευθύνσεις χωρίς να χάσουν χρόνο ή χρήμα. Αυτά τα χαρακτηριστικά εμφανίζονται σε πάρα πολλές παραγόμενες εξισώσεις με συχνότητα πέρα του 90%. Για το αρχείο Cocomo το χαρακτηριστικό LOC είναι αυτό που τόσο στις αριθμητικές όσο και στις λογικές εξισώσεις έχει συχνή εμφάνιση. Παράλληλα με αυτό στις αριθμητικές εκτελέσεις παρατηρήσαμε και το χαρακτηριστικό TIME να έχει πολλές εμφανίσεις. Από την άλλη, στις λογικές εξισώσεις είδαμε ακόμα δύο άλλα χαρακτηριστικά, το CPLX και το AEXP. Βλέποντας τώρα το δεύτερο αρχείο δεδομένων μας, παρατηρούμε μια ανάλογη εικόνα. Στις αριθμητικές εκτελέσεις υπάρχουν και πάλι δύο χαρακτηριστικά που σε αυτή την περίπτωση έχουν επιτυχία 100% στο σύνολο των καλύτερων εξισώσεων που έχουμε επιλέξει. Αυτά τα χαρακτηριστικά είναι το Length και το ENV. Στις εκτελέσεις των λογικών εξισώσεων βλέπουμε επίσης το χαρακτηριστικό PADJ να έχει συχνή εμφάνιση όπως και τα χαρακτηριστικά Length και TRANS. Όσο αφορά το χαρακτηριστικό LOC, η συχνή εμφάνιση του σε εξισώσεις δεν μας εξέπληξε εφόσον είναι ευρέως γνωστό ότι είναι ένας πάρα πολύ σημαντικός παράγοντας στον υπολογισμό του κόστους ανάπτυξης λογισμικού. Οι μεταβλητές αυτές είναι στο σύνολο τους μετρήσιμες μεταβλητές εξαιρουμένης της μεταβλητής LOC. Επομένως γνώση των τιμών των παραγόντων αυτών αποβαίνει χρήσιμη στα πρώτα στάδια της διαδικασίας ανάπτυξης λογισμικού και ένας υπολογισμός αρχικός είναι εφικτός.

Με τις πειραματικές διαδικασίες των λογικών εξισώσεων βγάζουμε ένα πολύ σημαντικό συμπέρασμα. Με αυτές τις εκτελέσεις έχουμε επιβεβαίωση των μεταβλητών που έχουμε μόλις αναφέρει ότι είναι σημαντικές. Ιδιαίτερα για τα αρχείο Cocomo, παρατηρήσαμε ότι η ίδια μεγάλη συχνότητα εμφανίσεων των χαρακτηριστικών LOC, CPLX, AEXP και TIME παρατηρείται και εδώ στα αποτελέσματα των εκτελέσεων μας. Για το αρχείο Desharnais, έχουμε επιβεβαίωση της μεταβλητής LENGTH, μερική επιβεβαίωση της μεταβλητής SCOPE, ενώ έχουμε επίσης μια άλλη μεταβλητή με συχνές εμφανίσεις, την μεταβλητή TRANS.

Κεφάλαιο 6

Συμπεράσματα και Εισηγήσεις για Μελλοντική Εργασία

6.1 Συμπεράσματα

6.2 Εισηγήσεις για μελλοντική εργασία

6.1 Συμπεράσματα

Μέσα από την εμπειρία αυτής της εφαρμογής έγινε πιο κατανοητή η σημασία αλλά και η μεγάλη προσπάθεια που γίνεται προκειμένου να γίνει σωστός υπολογισμός του κόστους ανάπτυξης λογισμικού. Είναι μια αρκετά επίπονη και χρονοβόρα διαδικασία η οποία πρέπει να γίνεται με πολύ προσεκτικό και μεθοδικό τρόπο. Ο σωστός υπολογισμός του κόστους μπορεί να αποβεί ακόμα και μοιραίος για την βιωσιμότητα του οργανισμού αν δεν γίνει σωστά, αλλά μπορεί να γίνει και χρυσός που θα του αποφέρει κέρδος οικονομικό αλλά και επιχειρησιακό. Μέχρι τώρα δεν ήταν κατορθωτό να βρεθεί μεθοδολογία η οποία ακολουθούμενη αυστηρά να μπορεί να δώσει ακριβή υπολογισμό κόστους αν και η έρευνες σε αυτό το τομέα συνεχίζονται ραγδαία. Με τις συνεχείς ανακαλύψεις που γίνονται στο τομέα αυτό μπορεί κανείς να πει ότι μάλλον είναι θέμα χρόνου να βρεθεί μια τέτοια πολύτιμη λύση.

Μέσα από το προτεινόμενο σύστημα προσπαθήσαμε να εξάξουμε κάποιους κανόνες οι οποίοι θα μπορούν να βοηθήσουν στην διαδικασία υπολογισμού του κόστους ανάπτυξης λογισμικού. Βασικό στοιχείο αποτέλεσε και αποτελεί η τιμή της προσπάθειας. Οι γενετικοί αλγόριθμοι είναι μια υποσχόμενη τεχνική η οποία μπορεί να αποδώσει καλά σε πολυδιάστατα προβλήματα χωρίς να χρειάζεται να γνωρίζει λεπτομέρειες για τη φύση του προβλήματος. Βάση αυτού στηριχθήκαμε και χρησιμοποιήσαμε στο σύστημα μας αυτή τη μεθοδολογία. Μέσω του συστήματος και κατ'επέκταση των γενετικών αλγορίθμων μπορέσαμε να κάνουμε κάποιες πειραματικές εκτελέσεις με χρήση αρχείων

που περιέχουν δεδομένα για ήδη υλοποιημένα έργα λογισμικού. Τα αποτελέσματα που μας έδωσαν αυτές οι εκτελέσεις ήταν κανόνες οι οποίοι κάποιες φορές έφτασαν ακόμα σύγκλιση 70%. Οι παραγόμενοι κανόνες είναι πολύ σημαντικοί εφόσον μπορούμε να εντοπίσουμε χαρακτηριστικά που είναι πολύ σημαντικά στην διαδικασία υπολογισμού του κόστους. Επιπρόσθετα χρήση των κανόνων αυτών από τους διαχειριστές του συστήματος πριν από την τελική διαδικασία υπολογισμού του κόστους μπορεί να είναι πολύτιμη εφόσον θα μπορεί να γίνεται ένας προκαταρτικός υπολογισμός που να μπορεί να δώσει κάποιες κατευθύνσεις στο πόσο να αναμένουν ότι θα είναι το κόστος υλοποίησης βάση κάποιων διαθέσιμων τιμών παραμέτρων που θα έχουν ήδη μπροστά τους. Με αυτό το τρόπο έργα τα οποία από τον πρώτο υπολογισμό μπορεί να δείξουν ότι έχουν πολύ ψηλό κόστος σε σχέση με το κέρδος που θα αποφέρουν, θα σταματούν και νέες κατευθύνσεις θα μπορούν να ακολουθηθούν χωρίς χάσιμο χρόνου ή ακόμα και κόστους.

Τα αποτελέσματα που πήραμε σε όλες τις ομάδες εκτελέσεων που έχουμε κάνει είναι ενδιαφέροντα. Κάποια μπορούν να θεωρηθούν ικανοποιητικά ενώ κάποια από αυτά χρήζουν περαιτέρω διερεύνησης. Αυτό δείχνει ότι η χρήση γενετικών αλγορίθμων σε προβλήματα της φύσης αυτής μπορεί να δώσει πάρα πολύ καλές λύσεις, επομένως περισσότερη έρευνα πάνω στον τομέα αυτό και βελτιώσεις στις διάφορες μεθόδους που ακολουθεί θα ήταν πολύ εποικοδομητικά. Επιπρόσθετα η μέτρηση της ακρίβειας δεν είναι απλή διαδικασία και πρέπει να χρησιμοποιηθούν ένας συνδυασμός από μέτρα προκειμένου να μπορέσουμε να έχουμε καλές μετρήσεις.

6.2 Εισηγήσεις για μελλοντική εργασία

Όπως δείχνουν και τα αποτελέσματα που πήραμε στις διάφορες εκτελέσεις η προτεινόμενη εφαρμογή είναι καλή και θα μπορούσε να χρησιμοποιηθεί για περαιτέρω μελέτη. Υπάρχουν κάποιες βελτιώσεις που θα μπορούσαν να γίνουν, όπως για παράδειγμα θα μπορούσε να γίνει μια έρευνα και κατά συνέπεια διόρθωση της συνάρτησης καταλληλότητας που επιχειρήσαμε να εισάγουμε η οποία δίνει τιμή ικανότητας σε μια λύση με βάση το συντελεστή MRE ή οποιοδήποτε από τους υπόλοιπους συντελεστές που υπολογίζουμε παράλληλα στο αρχείο εξόδου. Επίσης

μπορεί να γίνει μια έρευνα πάνω στις συναρτήσεις καταλληλότητας όχι μόνο για τη συγκεκριμένη συνάρτηση που αναφέραμε, αλλά γενικά να βρεθούν συναρτήσεις οι οποίες να βελτιστοποιούν τα αποτελέσματα μας στις αριθμητικές, λογικές και κατηγορικές εκτελέσεις.

Από τις εκτελέσεις μας δεν ήταν δυνατό λόγο χρόνου να έχουμε πολλές και διαφορετικές εκτελέσεις στο σύνολο δεδομένων ISBSG το οποίο είναι αρκετά μεγάλο και χρειάζεται με την παρούσα εφαρμογή 24 ώρες προκειμένου να τελειώσει μια εκτέλεση. Επομένως χρειάζεται να βρούμε μεγάλες σε δύναμη μηχανές και να εκτελέσουμε την εφαρμογή των κατηγορικών δεδομένων με αυτό το σύνολο για αρκετά πειράματα και να συγκρίνουμε και εκεί τα αποτελέσματα που θα μας δώσει.

Μελλοντική βελτίωση της εφαρμογής μπορεί να αποτελέσει και αλλαγή στις περιπτώσεις των λογικών εξισώσεων. Όπως έχουμε ήδη αναφέρει η εφαρμογή αυτή κάνει χρήση των ελάχιστων και μέγιστων τιμών κάθε μεταβλητής και με βάση αυτά δημιουργεί για κάθε μεταβλητή δύο τελεστέους με τους οποίους θα δουλέψει στην συνέχεια για να τρέξει ο γενετικός αλγόριθμος και να βρει κανόνες. Θα μπορούσαμε να αλλάξουμε την διαδικασία αυτή με τρόπο έτσι ώστε οι τελεστέοι να είναι οι μεταβλητές μας όπως έχουμε και στις αριθμητικές εξισώσεις και στο βήμα της δημιουργίας ενός νέου ατόμου στον πληθυσμό μας η εφαρμογή να επιλέγει εκτός από ένα τελεστή και ένα τελεστέο μια τυχαία τιμή η οποία όμως να βρίσκεται μέσα στο πεδίο της ελάχιστης και της μέγιστης τιμής του συγκεκριμένου τελεστέου. Δηλαδή να δημιουργούνται εξισώσεις της μορφής $X1 \text{ arg } A1$, όπου arg είναι ένας οποιοσδήποτε τελεστής από το σύνολο $\{<, >, =, <=, =>\}$, $X1$ μια μεταβλητή του αρχείου δεδομένων μας και $A1$ μια τυχαία τιμή η οποία είναι μεγαλύτερη ή ίση από την ελάχιστη τιμή της μεταβλητής $X1$ αλλά και μικρότερη ή ίση από την μέγιστη τιμή της μεταβλητής αυτής.

Μελλοντική εργασία αποτελεί επίσης και η απόδειξη ή η διάψευση της ένδειξης που πήραμε μέσω των διαφόρων εκτελέσεων μας ότι κάποια χαρακτηριστικά φαίνονται να είναι πάρα πολύ σημαντικά στην διαδικασία υπολογισμού του κόστους ανάπτυξης λογισμικού. Θα πρέπει να γίνει μια λεπτομερείς έρευνα αλλά και εύρεση τρόπων μέσω των οποίων να μπορεί να γίνει επαλήθευση η διάψευση της ένδειξης αυτής αναφορικά με τα χαρακτηριστικά LOC, TIME, CPLX και AEXP του αρχείου δεδομένων Cocomo και LENGTH, ENV, TRANS και PADJ του αρχείου δεδομένων Desharnais.

Σημαντική μελλοντική εργασία αποτελεί επίσης και η έρευνα των κατηγορικών εκτελέσεων. Η προσπάθεια που έχει γίνει σε αυτή τη μελέτη στον τομέα κατηγορικών εξισώσεων έγινε στηριζόμενοι σε κατηγορικές τιμές μεταβλητών. Επομένως μπορεί να εκτελεστούν σε αρχεία με μόνο κατηγορικές μεταβλητές. Ο στόχος είναι να έχουμε επεξεργασία αρχείων με κατηγορικές αλλά συνάμα και αριθμητικές μεταβλητές, όπου οι παραγόμενες εξισώσεις θα αποτελούν συνδυασμό των κατηγορικών συναρτήσεων με αριθμητικές συναρτήσεις.

Βιβλιογραφία

- [1] Adamopoulos, A.V.; Likothanassis, S.D. & Georgopoulos, E.F. (1998). A Feature Extractor of Seismic Data Using Genetic Algorithms, *Signal Processing IX: Theories and Applications, Proceedings of EUSIPCO-98, the 9th European Signal Processing Conference*, Vol. 2, pp. 2429-2432, Typporama, Greece.
- [2] Αγγελής Ελευθέριος, Μούτσα Μαρκέλλα (2004), "Εκτίμηση Κόστους Λογισμικού: Περιγραφή Αναδειγματοληπτικών τεχνικών για την μέθοδο των αναλογιών", 17^ο Πανελλήνιο Συνέδριο Στατιστικής
- [3] Aguilar-Ruiz S. Jesus, Ramos Isabel, Riquelme C.Jose, Toro Miguel, "An Evolutionary Approach to Estimating Software Development Projects", Universidad de Sevilla, Spain, 2001
- [4] Boehm, B.W. (1981). *Software Engineering Economics*, Prentice Hall, New Jersey.
- [5] Burgess, C.J. & Lefley, M. (2001). Can Genetic Programming Improve Software Effort Estimation? A Comparative Evaluation, *Information and Software Technology*, Vol. 43, No. 14, pp. 863-873, Elsevier, Amsterdam.
- [6] Center for Systems and Software Engineering, 20/11/2009, <http://csse.usc.edu/csse/research/COCOMOII/cocomo81.htm>
- [7] Dolado, J.J. (2001). On the Problem of the Software Cost Function, *Information and Software Technology*, Vol. 43, No. 1, pp. 61-72, Elsevier, Amsterdam.
- [8] Ευστράτιος Φ. Γεωργόπουλος, Σπυρίδων Δ. Λυκοθανάσης, "Εισαγωγή στους Γενετικούς Αλγόριθμους", Πολυτεχνική Σχολή Πανεπιστημίου Πατρών, 1999
- [9] "Genetic Algorithm", 11/01/2010, http://en.wikipedia.org/wiki/Genetic_Algorithm
- [10] "Genetic Programming", 20/12/2009, <http://www.genetic-programming.org/>
- [11] Holland, J.H. (1992). Genetic Algorithms, *Scientific American*, Vol. 267, No. 1, pp. 66–72, New York.
- [12] International Software Benchmarking Standards Group (ISBSG), Estimating, Benchmarking & Research Suite Release 9, ISBSG, Victoria, 2005.
- [13] International Software Benchmarking Standards Group, <http://www.isbsg.org/>
- [14] Ian Sommerville, "Software Engineering", eight ed., Addison-Wesley Longman Publishing Co., Inc., 2007
- [15] J.M. Desharnais, Analyse Statistique de la productivite des Projets de Development en Informatique a Partir de la Technique de Point de Fonction, Msc Thesis, Universite du Quebec, Montreal, 1989

- [16] Khoshgoftaar, T.M.; Evett, M.P.; Allen, E.B. & Chien, P. (1998). An Application of Genetic Programming to Software Quality Prediction *Computational Intelligence in Software Engineering, Series on Advances in Fuzzy Systems – Applications and Theory*, Vol. 16, pp. 176-195, World Scientific, Singapore.
- [17] Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Massachusetts.
- [18] Κώστας Κοντογιάννης, "Σημειώσεις Μαθήματος Τεχνολογίας Λογισμικού", Εθνικό Μετσόβιο Πολυτεχνείο, 2003
- [19] Lefley, M. & Shepperd, M.J. (2003). Using Genetic Programming to Improve Software Effort Estimation Based on General Data Sets, *Proceedings of GECCO*, pp. 2477-2487.
- [20] Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin.
- [21] Μιχαήλ Φ. Σταμούλης, "Επίλυση βέλτιστης ροής φορτίου σε Σ.Η.Ε υπό περιβαλλοντικούς περιορισμούς με την χρήση Γενετικών αλγορίθμων", Διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, 2009
- [22] Packard, N.H. (1990). A Genetic Learning Algorithm for the Analysis of Complex Data, *Complex Systems*, Vol. 4, No. 5, pp. 543-572, Illinois.
- [23] Π. Σ. Α. ΓΕΩΡΓΙΑΚΗΣ, "Εφαρμογή Γενετικών Αλγορίθμων στην Παραγωγή Ηλεκτρικής Ενέργειας", Πολυτεχνείο Κρήτης, Τεχν. Χρον. Επιστ. Έκδ. ΤΕΕ, ΙΙΙ, Τεύχ. 1-2
- [24] Sara Silva, "A genetic programming toolbox for Matlab", Version 3, ECOS - Evolutionary and Complex Systems Group University of Coimbra Portugal, 2007
- [25] Skinner M, "Genetic Algorithms Overview", 16/01/2010, <http://geneticalgorithms.ai-depot.com/Tutorial/Overview.html>
- [26] Sylvain Gelly, Olivier Teytaud, Nicolas Bredeche, Marc Schoenauer, "Universal Consistency and Bloat in GP", 2006
- [27] Σ. Λυκοθανάσης, Εισαγωγή στο Γενετικό Προγραμματισμό, Σημειώσεις Μαθήματος, Τμήμα Μηχανικών Η/Υ και Πληροφορικής
- [28] Σπυρίδων Α. Καζαρής, "Επίλυση Ωρολογίων Προγραμμάτων με Γενετικούς Αλγορίθμους", ΤΕΙ Σερρών
- [29] Y.F.Li, M. Xie, T.N. Goh, "A study of project selection and feature weightinh for analogy based software cost estimation", *The Journal of Systems and Software* 82 (2009) 241–252

Παράρτημα Α

Παρουσίαση Αποτελεσμάτων Εκτελέσεων

A.1 Εισαγωγή

A.2 Αποτελέσματα Αριθμητικών Εκτελέσεων – Αρχείο Cocomo

A.3 Αποτελέσματα Αριθμητικών Εκτελέσεων – Αρχείο Desharnais

A.4 Αποτελέσματα Λογικών Εκτελέσεων – Αρχείο Cocomo

A.5 Αποτελέσματα Λογικών Εκτελέσεων – Αρχείο Desharnais

A.6 Αποτελέσματα Κατηγορικών Εκτελέσεων – Αρχείο ISBSG

A.1 Εισαγωγή

Στόχος του παραρτήματος αυτού είναι να παραθέσουμε τα αποτελέσματα των εκτελέσεων που έχουμε κάνει κατά την πειραματική μας διαδικασία. Οι πίνακες που έχουν δημιουργηθεί είναι με βάση το μέγεθος του παραγόμενου δένδρου, ενώ έχουμε δώσει εδώ μια επιλογή καλύτερων αποτελεσμάτων καθώς οι εκτελέσεις που έχουν γίνει είναι πάρα πολλές και δεν μπορούν όλες να δοθούν στο παρόν αρχείο.

A.2 Αποτελέσματα Αριθμητικών Εκτελέσεων – Αρχείο Cocomo

Execution	A/A	Tree	Fitness	Total Fitness
Βάθος 4	25	$((ACAP+ACAP)*(LOC^*STOR))^*MODP$	11517.81693548	6021326.77141595
Βάθος 4	30	$((LOC^*TIME)^*TURN)*((VEXP+VEXP)^(PCAP+PCAP))$	8597.17452312	4738731.42338524
Βάθος 4	32	$((LOC^*TIME)^*MODP)+((LOC^*STOR)^*RELY)$	8045.08363887	4430168.96693229
Βάθος 4	65	$((LOC/MODP)^*STOR)+((LOC^*TIME)^*MODP)$	6640.94451074	3777279.15288114
Βάθος 4	74	$((LOC^*RELY)*(STOR^*RELY))+((LOC^*TIME)^*MODP)$	7354.21364033	3992457.30392057
Βάθος 4	117	$((LOC^*SCED)^*(TIME/TOOL))+((LOC^*VEXP)^*(TIME^*MODP))$	8661.06460208	5198816.22119926
Βάθος 4	137	$((LOC/VEXP)^(TIME/TOOL))+((LOC^*TIME)^*(VEXP^*MODP))$	8315.77291634	4544117.70199209

Πίνακας A.1α: Αριθμητικά Cocomo A - Βάθος 4 Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Βάθος 4	25	0.246271	0.973067	201973.373451	0.538291	225.839547	0.156862	1.327977	0.518644	0.966136	955786.125678	0.562427	335.125927	0.333333	1.536637
Βάθος 4	30	0.199635	0.980867	135265.039491	0.553568	168.572049	0.431372	0.060343	0.380927	0.921825	469843.658642	0.592137	373.706606	0.083333	-0.158786
Βάθος 4	32	0.182991	0.985818	109861.699708	0.449675	157.746738	0.352941	0.973840	0.349120	0.962136	457721.192070	0.497039	313.170731	0.166666	1.375784
Βάθος 4	65	0.160808	0.988441	86364.262801	0.463590	130.214598	0.235294	0.701984	0.561984	0.988188	1079591.486644	0.543898	538.855972	0.083333	1.857127
Βάθος 4	74	0.174760	0.986882	100264.998556	0.428754	144.200267	0.352941	0.971508	0.366884	0.959564	498507.171966	0.585957	370.743231	0.166666	1.385692
Βάθος 4	117	0.183567	0.985169	135470.812861	0.459362	169.824796	0.333333	1.011420	0.322471	0.945445	20095.173384	0.468547	81.411789	0.333333	0.082409
Βάθος 4	137	0.180437	0.985132	131898.472909	0.468259	163.054370	0.333333	0.812681	0.559880	0.884320	31970.999515	0.537339	119.726628	0.25	1.215904

Πίνακας Α.16: Αριθμητικά Cosmo A - Βάθος 4 Τιμές

Execution	A/A	Tree	Fitness	Total Fitness
Βάθος 5	83	(((LOC^AEXP)^DATA)+((TOOL*LOC)^MODP))^STOR)	7913.81736981	4805096.99268285
Βάθος 5	103	(((LOC^MODP)+(STOR+STOR))^TIME)+(((SCED+LOC)^RELY)*(STOR^RELY))	7551.02162356	4070766.41032466
Βάθος 5	110	(((LOC*LEXP)^RELY)+((LOC+LOC)^(ACAP*VIRT)))+(((LOC^MODP)*VIRT)^TIME))	6336.04062632	3914881.45247241
Βάθος 5	112	(((LOC*MODP)*AEXP)^TIME)*(((STOR*SCED)+(SCED+DATA))^(CPLX*(PCAP*SCED))))	4280.82133191	2905074.28490863
Βάθος 5	121	(((ACAP+ACAP)*((MODP+MODP)+(LOC^TIME)))^DATA)	12048.07116998	6550036.69404778
Βάθος 5	6	(((LOC^STOR)+(LOC^AEXP))*((TIME^ACAP)^TIME))^MODP)	7782.11437286	4478704.68139259
Βάθος 5	11	(((LOC^(STOR*MODP))+((TIME+TIME)^(VEXP+STOR)))+((LOC^(TIME*DATA))^AEXP))	6401.74550936	3668276.62979903
Βάθος 5	37	(((VIRT*LOC)^(MODP*CPLX)*(MODP*CPLX))*(((STOR+LOC)+(STOR+VIRT))^TIME))	9586.97691602	5843218.9996349
Βάθος 5	124	(((LOC+CPLX)^RELY)*VEXP)+(((LOC^MODP)^TIME)+((LOC^DATA)*CPLX))	8661.03501291	5126047.85519936
Βάθος 5	128	(((AEXP/(PCAP/TIME))*((RELY*LOC)+(LOC^STOR)))^MODP)	12351.61016705	7014681.8278364

Πίνακας Α.2α: Αριθμητικά CoCoMo A - Βάθος 5 Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Βάθος 5	83	0.135075	0.99161	73125.288376	0.467068	155.172889	0.274509	0.671206	0.312808	0.963617	20433.530135	0.515984	87.198467	0.333333	1.113162
Βάθος 5	103	0.179082	0.986034	105430.773407	0.465161	148.059247	0.392156	0.630920	0.353700	0.962130	463992.891230	0.497102	338.835969	0.166666	1.212438
Βάθος 5	110	0.176154	0.985472	104285.866180	0.429821	124.236090	0.411764	0.131709	0.367278	0.958549	455698.395628	0.552580	340.173331	0.166666	0.465135
Βάθος 5	112	0.106123	0.994442	37665.822133	0.447366	83.937673	0.333333	-0.046378	0.515189	0.955250	891128.347192	0.517833	516.513030	0.083333	0.682625
Βάθος 5	121	0.378665	0.928760	577935.815837	0.482475	236.236689	0.294117	0.895575	0.564997	0.855426	94002.719758	0.552177	177.041477	0.083333	1.658622
Βάθος 5	6	0.191314	0.983522	121643.377038	0.476633	152.590477	0.235294	1.168804	0.597460	0.974124	1262512.247742	0.518269	468.177428	0.166666	1.506100
Βάθος 5	11	0.138218	0.991288	63670.176505	0.484581	125.524421	0.313725	0.352811	0.242901	0.978594	205973.586974	0.493973	249.402638	0.25	1.199231
Βάθος 5	37	0.201458	0.982037	163551.394364	0.512082	187.979939	0.333333	2.160979	0.514393	0.929751	43967.820457	0.560613	119.692891	0.083333	2.203822
Βάθος 5	124	0.183762	0.985720	136434.214345	0.540067	169.824215	0.294117	0.278886	0.581316	0.875966	56221.964231	0.545737	134.860919	0.25	0.508985
Βάθος 5	128	0.319525	0.953703	412485.661368	0.504676	242.188434	0.235294	1.138857	0.567817	0.872237	36158.060449	0.539492	124.099944	0.083333	0.979548

Πίνακας Α.2β: Αριθμητικά Cocoto A - Βάθος 5 Τιμές

Βλέποντας τα πιο πάνω αποτελέσματα των εκτελέσεων με βάση το βάθος του δένδρου, έχουμε τονίσει με μαύρο χρώμα τα αποτελέσματα που θεωρούνται ως τα καλύτερα. Για το MRE έχουμε μικρότερη τιμή 0.468547 στην εκτέλεση 117 βάθους 4, όπου στην ίδια εκτέλεση έχουμε και καλύτερη τιμή του BRE 0.082409 αλλά και μεγαλύτερη τιμή που πήραμε στο PRED 0.333333. Όσο αφορά το CC έχουμε μεγαλύτερη τιμή 0.988188 στο 65 του βάθους 4 ενώ για το NRMSE έχουμε 0.242901 στο 11 του βάθους 5.

Execution	A/A	Tree	Fitness	Total Fitness
Κόμβοι 16	34	$(((((\text{SCED} * \text{LOC})^{\text{RELY}}) + ((\text{LOC}^{\text{TIME}} / \text{STOR})^{\text{MODP}}) * \text{STOR}))$	4998.97215205	2856639.06396801
Κόμβοι 16	35	$((\text{LOC}^{\text{STOR}}) + ((\text{TIME} + ((\text{LOC}^{\text{MODP}} + \text{TIME}))^{\text{TIME}}) + \text{TIME}))$	7955.3028256	4215436.93087329
Κόμβοι 16	111	$(((((\text{LOC} + (\text{LOC}^{\text{RELY}})^{\text{DATA}})^{\text{TOOL}}) + (\text{LOC}^{\text{TIME}}))^{\text{DATA}})$	9395.02392391	5512182.10745651
Κόμβοι 16	113	$(((((\text{LOC}^{\text{TURN}})^{\text{CPLX}}) + ((\text{LOC}^{\text{TIME}}) + (\text{LOC}^{\text{ACAP}})))^{\text{DATA}})$	7228.09631871	4412855.09919203
Κόμβοι 16	144	$(((((\text{LOC} * \text{RELY})^{\text{CPLX}}) + ((\text{MODP} * (\text{STOR} + \text{LOC}))^{\text{TIME}}))^{\text{DATA}})$	9093.74807136	5033518.26413112

Πίνακας Α.3α: Αριθμητικά Cosomo A – Κόμβοι 16 Παραγόμενες Εξιιώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Κόμβοι 16	34	0.113447	0.994467	41871.049727	0.426462	98.019061	0.333333	0.900992	0.458122	0.939225	747076.851323	0.548091	538.198118	0.25	1.461107
Κόμβοι 16	35	0.171866	0.987200	98219.501311	0.471415	155.986329	0.235294	0.650128	0.512023	0.982450	933731.684848	0.596711	439.736246	0.333333	0.686656
Κόμβοι 16	111	0.192918	0.981467	147792.182593	0.528503	184.216155	0.352941	0.564588	0.547692	0.932144	141851.294884	0.558972	176.427952	0.25	-0.281301
Κόμβοι 16	113	0.130848	0.991806	68933.508323	0.539515	141.727378	0.294117	0.443210	0.417546	0.916094	22892.356659	0.570831	92.382511	0.333333	0.789067
Κόμβοι 16	144	0.174977	0.987288	123266.074396	0.476661	178.308785	0.352941	0.886180	0.470126	0.930870	5343.652725	0.493509	61.199005	0.166666	1.051716

Πίνακας Α.3β: Αριθμητικά Cosomo A - Κόμβοι 16 Τιμές

Execution	A/A	Tree	Fitness	Total Fitness
Κόμβοι 20	12	$(((((LOC^{(MODP*TIME)})/(MODP/SCED)))/(AEXP/SCED))+((LOC^{STOR}/PCAP)))$	6670.7404633	3579402.09099942
Κόμβοι 20	78	$(((((LOC^{DATA})+CPLX)^{AEXP})^{ACAP})+(((LOC^{SCED})^{SCED})^{(STOR*DATA)}))$	4766.83087377	2568789.37547136
Κόμβοι 20	89	$(TIME+(((LOC^{DATA})^{CPLX})+(\log((ACAP^{LOC}))+(LOC^{STOR})))^{VEXP})^{SCED})$	6249.16285204	3399141.80420654

Πίνακας Α.4α: Αριθμητικά CoCoMo Α – Κόμβοι 20 Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Κόμβοι 20	12	0.160431	0.988017	85592.261362	0.514811	130.798832	0.313725	0.668906	0.487766	0.961615	832297.178419	0.522688	449.419888	0.25	0.834971
Κόμβοι 20	78	0.361574	0.946184	36111.864833	0.440042	93.467272	0.313725	0.917445	0.469120	0.878807	3165995.990548	0.491500	985.9646419	0.25	1.375678
Κόμβοι 20	89	0.293325	0.960636	87784.964303	0.418647	122.532604	0.392156	0.554242	0.468026	0.886756	2737067.285630	0.588312	773.847869	0.083333	0.803440

Πίνακας Α.4β: Αριθμητικά CoCoMo Α - Κόμβοι 20 Τιμές

Execution	A/A	Tree	Fitness	Total Fitness
Κόμβοι 28	29	((((((LOC/(((TOOL^CPLX)^CPLX)^CPLX)^CPLX)/TIME)/VEXP)/TURN)^STOR)+(((LOC^VEXP)^MODP)^TIME))	7910.82181484	4424050.20687536
Κόμβοι 28	42	((((((((((LOC^STOR)+TIME)+TIME)+TIME)*PCAP)+TIME)+TIME)*SCED)/MODP)+(((LOC^TIME)*VIRT)^MODP))	7823.00600345	7612493.9871253
Κόμβοι 28	83	(((TIME*(LOC^STOR))^MODP)-(((MODP)^((VEXP*(LOC*(PCAP*STOR)))^VIRT))))-log(((TIME)^LOC))	5438.52355294	2999270.9420341
Κόμβοι 28	107	(LOC+((((((((((LOC^(DATA*TIME))+LOC)+LOC)^VEXP)*PCAP)*VEXP)*AEXP)*PCAP)*AEXP)/DATA)^SCED))	4055.37612813	2819048.6338639
Κόμβοι 28	113	(((TIME^TIME)*LOC)^((MODP)+VIRT))+((ACAP*TIME)*LOC)+((CPLX*(STOR+LOC))^(DATA+TIME)))	7478.29770588	4134250.65887925
Κόμβοι 28	123	((((((LOC)*CPLX)*(((CPLX+CPLX)+LOC)^TIME)+LOC)+(STOR+(STOR+STOR))))*VEXP)*PCAP)*VEXP)^TURN)	8443.613497	4496250.6857475
Κόμβοι 28	150	((VEXP*((SCED*STOR)*(TIME+((TIME+(((LOC^(log(SCED)+MODP))*AEXP)^DATA))*SCED)))+(LOC^(SCED*STOR)))	6239.57012985	3570178.19150735

Πίνακας Α.5α: Αριθμητικά CoCoMo Α - Κόμβοι 28 Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Κόμβοι 28	29	0.173542	0.986532	121661.249728	0.462427	155.114153	0.352941	0.812034	0.529143	0.923076	26914.385481	0.494081	116.123672	0.166666	1.546791
Κόμβοι 28	42	0.175751	0.986630	103388.582347	0.484549	153.392274	0.313725	0.561018	0.439155	0.984200	671220.309813	0.520476	349.452908	0.166666	0.578077
Κόμβοι 28	83	0.249945	0.970462	62555.219607	0.432355	106.637716	0.372549	0.131955	0.380291	0.965593	1872071.574808	0.548285	593.820418	0.25	0.291039
Κόμβοι 28	107	0.086510	0.996509	24949.444078	0.454399	79.517178	0.450980	0.023736	0.326529	0.972617	354918.885432	0.599758	397.889611	0.333333	0.495014
Κόμβοι 28	113	0.137588	0.990752	75922.820510	0.505519	146.633288	0.352941	0.216892	0.165555	0.987951	8115.734057	0.547140	49.562412	0.5	0.078463
Κόμβοι 28	123	0.196668	0.980745	128102.306220	0.496294	165.561048	0.313725	0.717449	0.205709	0.980242	153196.118914	0.596647	173.397095	0.416666	0.484809
Κόμβοι 28	150	0.125634	0.992950	52490.663957	0.473216	122.344512	0.254901	0.625344	0.248443	0.973507	219951.274575	0.503797	208.037396	0.333333	0.028365

Πίνακας Α.5β: Αριθμητικά CoCoMo Α - Κόμβοι 28 Τιμές

Execution	A/A	Tree	Fitness	Total Fitness
Βάθος 4	8	$((LOC*STOR)^{RELY})+((LOC^{TIME})^{MODP})$	4339.73564213	1863169.67666062
Βάθος 4	19	$((LOC^{TIME})^{MODP})+((LOC^{RELY})^{CPLX})$	5083.00022302	2341147.36886564
Βάθος 4	49	$((LOC^{STOR})^{MODP})+((LOC^{STOR})+(LOC^{AEXP}))$	6247.76951128	2423838.25259321
Βάθος 4	58	$((LOC^{TIME})^{MODP})+((LOC/MODP)^{STOR})$	5230.42593428	2094815.08441719
Βάθος 4	77	$((CPLX*(LOC^{RELY}))+(LOC+STOR)^{(MODP*TIME)})$	6081.95717276	2354367.1457082
Βάθος 4	79	$((LOC^{STOR})/TURN)+((LOC^{TIME})^{MODP})$	4047.54880869	1813202.29614908
Βάθος 4	98	$((LOC^{TURN})^{(TIME*SCED)})+((LOC^{VEXP})^{(TIME*DATA)})$	6385.38600318	2668637.18219658

Πίνακας Α.6α: Αριθμητικά Cocomo Β - Βάθος 4 Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Βάθος 4	8	0.096224	0.996708	39519.023776	0.449591	114.203569	0.289473	0.909563	0.439086	0.933000	383755.123459	0.472538	298.535871	0.36	1.264473
Βάθος 4	19	0.118444	0.994380	60210.925982	0.438236	133.763163	0.315789	0.924790	0.490548	0.945067	467985.597898	0.512655	319.115649	0.08	1.369064
Βάθος 4	49	0.316356	0.956247	126127.192818	0.564873	164.414987	0.368421	0.540087	0.362152	0.938558	844107.008107	0.594747	348.318868	0.36	0.000876
Βάθος 4	58	0.154899	0.989280	106087.210411	0.478186	137.642787	0.210526	0.688287	0.553802	0.964112	533134.449878	0.479952	315.071610	0.2	1.277273
Βάθος 4	77	0.166891	0.988237	120988.381278	0.458703	160.051504	0.263157	0.913446	0.447111	0.958253	373294.192247	0.470037	276.258123	0.24	1.037581
Βάθος 4	79	0.093907	0.996571	38468.546136	0.445347	106.514442	0.263157	0.822907	0.567895	0.943995	594946.823717	0.525366	354.387661	0.2	1.019644
Βάθος 4	98	0.157678	0.988832	109528.510097	0.444688	168.036473	0.289473	0.805488	0.570505	0.851976	575082.588223	0.582191	342.573841	0.12	0.740747

Πίνακας Α.6β: Αριθμητικά Cocomo Β - Βάθος 4 Τιμές

Execution	A/A	Tree	Fitness	Total Fitness
Βάθος 5	2	$((DATA^{TIME}) * ((LOC^{MODP})^{STOR})) + (((LOC^{AEXP}) + (LOC - STOR))^{TIME}))$	3162.90489143	1626199.46887129
Βάθος 5	29	$(((((LOC^{DATA})^{(AEXP/LEXP)}) + ((LOC^{SCED})^{(LEXP^{CPLX}))})^{TIME}))$	5799.32950317	2241387.14852615
Βάθος 5	32	$(((((LOC^{SCED})^{STOR}) + ((LOC^{SCED})^{(TURN^{MODP}))})^{*STOR}))$	3390.365338	1377200.34526547
Βάθος 5	39	$(((((LOC^{TIME})^{PCAP})^{CPLX}) + (((LOC^{TIME}) + STOR)^{DATA}))$	5021.14556356	2493363.4633135
Βάθος 5	87	$(((((LOC^{DATA})^{(TURN^{PCAP})}) + ((LOC^{PCAP})^{CPLX}))^{TIME}))$	6662.72541917	3101464.78269217
Βάθος 5	92	$(((((LOC^{STOR})^{MODP})/TOOL) + ((LOC^{(DATA^{TIME})}) + ((TIME + TIME)^{(TIME + TIME)})))$	5965.62670892	2309399.11841683

Πίνακας Α.7α: Αριθμητικά Cosomo B - Βάθος 5 Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Βάθος 5	2	0.065482	0.997821	22288.744031	0.501014	83.234339	0.342105	0.229549	0.595039	0.848337	172508.949694	0.595931	219.628294	0.36	0.061424
Βάθος 5	29	0.180583	0.985773	115193.568425	0.452910	152.613934	0.315789	0.662576	0.560733	0.867284	982010.943877	0.510281	387.930814	0.32	0.764912
Βάθος 5	32	0.180389	0.984601	39068.108965	0.446864	89.220140	0.289473	0.560142	0.391579	0.955147	984853.724530	0.565781	397.643256	0.16	0.792929
Βάθος 5	39	0.101705	0.995695	45324.004531	0.427264	132.135409	0.315789	0.502480	0.404158	0.951344	288580.635442	0.501550	234.817689	0.2	1.162959
Βάθος 5	87	0.161259	0.987752	112661.330768	0.483276	175.334879	0.368421	0.826495	0.401724	0.968781	292032.207428	0.518099	210.772772	0.32	0.612360
Βάθος 5	92	0.143610	0.990431	89171.651149	0.488182	156.990176	0.236842	0.462659	0.242058	0.974087	109015.797585	0.563166	156.644487	0.24	0.495765

Πίνακας Α.7β: Αριθμητικά Cosomo B - Βάθος 5 Τιμές

Execution	A/A	Tree	Fitness	Total Fitness
Βάθος 6	9	$(((LOC^{\wedge}STOR)^{\wedge}VIRT)^{\wedge}TOOL)^{\wedge}AEXP)+(((LOC^{\wedge}DATA)^{\wedge}AEXP)^{\wedge}CPLX)+(((LOC^{\wedge}VEXP)^{\wedge}SCED)^{\wedge}STOR))$	2318.81255342	1377398.77676715
Βάθος 6	34	$(((VIRT+VIRT)^{\wedge}(VIRT+VIRT))+LOC^{\wedge}MODP)^{\wedge}TIME)+(((LOC+(VIRT^{\wedge}CPLX))^{\wedge}STOR)/MODP)$	5168.34467322	1950940.26719104
Βάθος 6	51	$((RELY^{\wedge}((LOC^{\wedge}SCED)^{\wedge}STOR))+((LOC^{\wedge}TURN)^{\wedge}TURN)^{\wedge}TURN)+(((LOC^{\wedge}VEXP)^{\wedge}STOR)^{\wedge}MODP)^{\wedge}AEXP))$	1998.68962478	945370.99021587
Βάθος 6	68	$((VIRT^{\wedge}((LOC+(VIRT/RELY))^{\wedge}(MODP^{\wedge}(TIME^{\wedge}VIRT)))))+(((LOC^{\wedge}PCAP)+(PCAP+RELY))/((TIME^{\wedge}MODP)+CPLX))+((LOC+ACAP)^{\wedge}RELY)))$	3720.01772365	1622744.80006252

Πίνακας Α.8α: Αριθμητικά Cosoto B - Βάθος 6 Παραγόμενες Εξιιώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Βάθος 6	9	0.081846	0.997160	8870.277417	0.495159	61.021382	0.289473	0.042043	0.445052	0.972581	1252296.675285	0.501298	434.543076	0.36	0.275398
Βάθος 6	34	0.154276	0.988987	105180.916086	0.479582	136.009070	0.368421	0.269107	0.538042	0.967415	506430.869848	0.582602	299.563710	0.16	0.642363
Βάθος 6	51	0.074931	0.997142	7119.083351	0.486791	52.597095	0.447368	0.077736	0.563093	0.970769	2034899.416834	0.520186	533.222656	0.24	0.691023
Βάθος 6	68	0.103210	0.995535	46068.998263	0.429370	97.895203	0.394736	0.331296	0.446264	0.938833	379132.036565	0.499686	292.333556	0.2	0.884549

Πίνακας Α.8β: Αριθμητικά Cosoto B - Βάθος 6 Τιμές

Execution	A/A	Tree	Fitness	Total Fitness
Βάθος 7	27	$(((((LOC^{AEXP})^{ACAP})+(LOC/STOR)^{RELY})^{DATA})^{PCAP})+(((LOC^{STOR})^{TOOL})^{TOOL})+(TIME^{STOR})^{TOOL})^{RELY})$	3769.93237699	1477178.7894503
Βάθος 7	65	$(((((LOC+VIRT)^{VIRT})-RELY)^{SCED})^{STOR})+(((LOC-STOR)-RELY)^{STOR})^{MODP})$	4138.281685	1627848.26959554
Βάθος 7	77	$((VEXP^{(VEXP^{(PCAP+STOR)})})^{((LOC/(VEXP+RELY))+LOC)^{TIME})^{TURN})$	3445.57839764	1451114.87957835
Βάθος 7	84	$((LOC^{TIME})+(((LOC^{TIME})^{TIME})^{ACAP})+(((LOC^{PCAP})-CPLX)^{CPLX})^{DATA})$	3314.3863273	1409345.67727743

Πίνακας A.9α: Αριθμητικά Cocomo B - Βάθος 7 Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Βάθος 7	27	0.212976	0.978884	53072.264049	0.352763	99.208746	0.5	0.488249	0.585079	0.976949	2229707.341373	0.527518	590.333661	0.16	0.802368
Βάθος 7	65	0.176755	0.985075	67311.390603	0.463306	108.902149	0.315789	0.664920	0.441045	0.983988	1013620.156492	0.538116	401.059152	0.16	1.723032
Βάθος 7	77	0.093874	0.995737	30051.915729	0.521258	90.673115	0.421052	0.232178	0.518269	0.864309	872225.006873	0.587132	468.281310	0.32	0.275346
Βάθος 7	84	0.151513	0.988975	28183.996656	0.455617	87.220692	0.342105	0.431441	0.397842	0.983832	1026832.410661	0.565094	393.385201	0.32	0.703715

Πίνακας A.9β: Αριθμητικά Cocomo B - Βάθος 7 Τιμές

Με τις εκτελέσεις με κριτήριο το βάθος του δένδρου βλέπουμε ότι καλύτερη τιμή για MRE 0.470037 έχουμε στην εκτέλεση 77 του βάθους 4, για CC έχουμε 0.983988 στην εκτέλεση 65 με βάθος 7, ενώ για NRMSE έχουμε 0.242058 στην εκτέλεση 92 του βάθους 5. Για τις τιμές των δύο άλλων συντελεστών, έχουμε ότι για το BRE η τιμή 0.000876 της εκτέλεσης 49 του βάθους 4 είναι πάρα πολύ καλή αφού σχεδόν έχουμε τιμή 0

που θεωρείται ως η ιδανική τιμή για αυτό το συντελεστή. Από την άλλη για το συντελεστή PRED, και πάλι δεν έχουμε τιμή να ξεπερνά το 0.6 με καλύτερη τιμή το 0.36 σε 2 με 3 εκτελέσεις.

Execution	A/A	Tree	Fitness	Total Fitness
Κόμβοι 16	5	$(((((TIME*(TIME*(LOC^STOR)))+TIME)+TIME)^{MODP})+LOC)$	5232.90604151	2083651.99970751
Κόμβοι 16	30	$((VIRT*(LOC^STOR))+((TOOL^ACAP)*((LOC^STOR)^{MODP})))$	3505.60365251	1366743.28629296
Κόμβοι 16	37	$(MODP*(AEXP*(AEXP*((LOC*((MODP+AEXP)+MODP))^TIME))))$	4682.17206287	1882420.65768273
Κόμβοι 16	66	$((TIME+(AEXP*LOC))+(((LOC*TIME)^{STOR*MODP})*TIME))$	4249.30754773	1814163.5577568
Κόμβοι 16	67	$(LOC^(((MODP+(((ACAP+STOR)^{MODP})+DATA))*STOR)))$	6443.00464911	2793901.81086144
Κόμβοι 16	81	$(((((LOC^SCED)*AEXP)/PCAP)^{STOR})+((LOC^STOR)^{MODP}))$	4887.8617617	1861349.17271643
Κόμβοι 16	94	$(((((LOC^VEXP)^TIME)^{MODP})+LOC)+((LOC^STOR)^RELY))$	6555.9181448	2599405.36303024

Πίνακας A.10α: Αριθμητικά Cosoto B - Κόμβοι 16 Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Κόμβοι 16	5	0.115195	0.994216	56866.196502	0.450594	137.708053	0.342105	0.900266	0.513474	0.950316	516154.148668	0.541209	314.611324	0.28	0.589945
Κόμβοι 16	30	0.118594	0.993440	29719.113415	0.451017	92.252727	0.289473	0.748897	0.417227	0.982840	914195.063760	0.551697	415.943861	0.12	1.066363
Κόμβοι 16	37	1.066363	0.988376	82283.292932	0.480111	123.215054	0.342105	0.208340	0.563048	0.862404	1012666.895906	0.536620	372.734197	0.36	0.402118
Κόμβοι 16	66	0.102867	0.995349	45230.792678	0.435820	111.823882	0.342105	0.788760	0.522707	0.928637	541181.638966	0.526561	349.496298	0.28	0.872778
Κόμβοι 16	67	0.179773	0.985373	140839.745667	0.549167	169.552753	0.210526	3.292817	0.461021	0.974687	390330.166578	0.597925	280.105652	0.16	2.777046
Κόμβοι 16	81	0.158292	0.989125	53108.451742	0.451216	128.627941	0.342105	0.743097	0.378304	0.986934	757236.820863	0.559125	323.983539	0.24	0.441763
Κόμβοι 16	94	0.156844	0.989508	107351.490267	0.477755	172.524161	0.342105	0.213618	0.566857	0.955103	573436.457400	0.553727	263.490184	0.32	0.387160

Πίνακας Α.10β: Αριθμητικά Cosmo B - Κόμβοι 16 Τιμές

Execution	A/A	Tree	Fitness	Total Fitness
Κόμβοι 20	4	$((((LOC^{DATA})^{RELY})+(LOC^{DATA}))*((ACAP+SCED))*((CPLX*VEXP)*ACAP)))$	5475.69594688	2075855.77208909
Κόμβοι 20	68	$(LOC+((TIME*(((LOC^{TIME})^{TIME})^{TIME})^{PCAP})^{SCED}))*((LOC^{STOR})))$	2427.2030445	957992.09306096
Κόμβοι 20	70	$(((((LOC^{TOOL})^{MODP})+(((LOC-STOR)^{AEXP})/MODP))*DATA)*DATA)^{STOR}$	6830.78921368	2950448.53837233

Πίνακας Α.11α: Αριθμητικά CoCoMoB - Κόμβοι 20 Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Κόμβοι 20	4	0.527362	0.867468	97435.568587	0.500320	144.097261	0.342105	0.511266	0.579067	0.844172	2603386.599513	0.539413	601.295272	0.32	-0.019096
Κόμβοι 20	68	0.094156	0.996142	10341.957725	0.462305	63.873764	0.342105	0.310394	0.557775	0.934782	2014307.229879	0.565979	658.989494	0.16	1.252001
Κόμβοι 20	70	0.140649	0.990957	103857.256411	0.463319	179.757610	0.315789	0.935020	0.507954	0.885455	37253.920601	0.495911	101.470403	0.32	0.985140

Πίνακας Α.11β: Αριθμητικά CoCoMo B - Κόμβοι 20 Τιμές

Execution	A/A	Tree	Fitness	Total Fitness
Κόμβοι 28	26	$(VIRT+(VIRT+(((LOC^*STOR)^MODP)+(((LOC)-LOC)+(VEXP*(VEXP*(SCED*((LOC*(DATA+MODP))^TIME))))))))$	4555.74822505	1990440.66715849
Κόμβοι 28	47	$((((LOC^*TIME)^DATA)-(DATA*(DATA/(((VIRT^*RELY)^RELY)^RELY)^LOC))))+((LOC^*STOR)^MODP))$	4720.77833767	9999999.99999999
Κόμβοι 28	54	$((TIME^(((MODP*(LOC))+((LOC^*PCAP)*TOOL))^VEXP))^(((LOC^*PCAP)*TURN)+((LOC^*CPLX)*DATA))$	6433.69295603	2832565.3169383
Κόμβοι 28	88	$((LOC^(PCAP*(VEXP*(TIME*DATA))))+(((AEXP^*TIME)^TIME)^(TIME/VEXP)^*TIME)*(LOC^(CPLX*TIME)))$	5734.57722196	2754089.27682843
Κόμβοι 28	92	$((((LOC-MODP)^*STOR)^MODP)+(((((((LOC-MODP)^MODP)^TIME)+STOR)+STOR)/MODP)+STOR)/MODP)/MODP))$	2854.33663055	1232976.95977243

Πίνακας A.12α: Αριθμητικά Cosmo B - Κόμβοι 28 Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Κόμβοι 28	26	0.126601	0.992023	70524.594814	0.476787	119.888111	0.421052	0.053115	0.326028	0.948520	187654.878594	0.481071	204.036352	0.16	0.375912
Κόμβοι 28	47	0.114502	0.993801	68476.842806	0.445578	124.231008	0.289473	0.657779	0.593850	0.843162	132311.772173	0.471871	183.510692	0.08	0.975528
Κόμβοι 28	54	0.173591	0.986159	158544.937711	0.427614	169.307709	0.368421	0.788309	0.562022	0.872312	22323.473899	0.481968	91.351973	0.32	0.354810
Κόμβοι 28	88	0.120669	0.994067	75872.553250	0.410233	150.909926	0.315789	0.778826	0.545392	0.880246	74440.938887	0.519237	139.958491	0.2	0.570406
Κόμβοι 28	92	0.062762	0.998239	16895.068521	0.457373	75.114121	0.289473	0.637333	0.536191	0.939455	562895.871448	0.585337	387.415864	0.08	1.275822

Πίνακας A.12β: Αριθμητικά Cosmo B - Κόμβοι 28 Τιμές

A.3 Αποτελέσματα Αριθμητικών Εκτελέσεων – Αρχείο Desharnais

Execution	A/A	Tree	Fitness	Total Fitness
Βάθος 5	86	$(((((\text{DURATION MONTHS} * \text{DURATION MONTHS}) + \text{SCOPE}) + ((\text{MANAGER EXP.} + \text{POINTS AJUST.}) + (\text{TRANSACTIONS} + \text{MANAGER EXP.}))) * ((\text{POINTS NON AJUST.} - (\text{TRANSACTIONS} + \text{DURATION MONTHS}))))))$	106377.85800109	56599284.9208746
Βάθος 6	22	$(((((\text{DURATION MONTHS} * \text{ENTITIES}) + (\text{ENTITIES} - \text{POINTS AJUST.})) * ((\text{SCOPE} * \text{TRANSACTIONS}) / (\text{ENTITIES} - \text{TRANSACTIONS})))) * \text{POINTS AJUST.})$	112423.66031733	57255116.7712313
Βάθος 7	51	$(((((\text{DURATION MONTHS} * \text{SCOPE}) * ((\text{SCOPE} - (\text{ENTITIES})) * \text{DURATION MONTHS}))) * \text{POINTS NON AJUST.})$	108734.28327081	54867820.576642
Βάθος 7	55	$(((((\text{TRANSACTIONS} * \text{TRANSACTIONS}) * (\text{TRANSACTIONS} * \text{TRANSACTIONS})) * ((\text{SCOPE} ^ \text{DURATION MONTHS}) * (\text{SCOPE} - \text{DURATION MONTHS}))) ^ \text{MANAGER EXP.})) + (((\text{POINTS AJUST.} + \text{POINTS AJUST.}) + (\text{SCOPE} * \text{DURATION MONTHS})) * (\text{SCOPE}))$	111789.19729138	56770108.5979203
Βάθος 7	65	$((((\log((\text{POINTS NON AJUST.} * \text{TRANSACTIONS})) * \log(((\text{ENTITIES} / \text{SCOPE}) * \text{TRANSACTIONS})))) * (((\text{POINTS AJUST.} + (\text{DURATION MONTHS} * \text{SCOPE})) + (\text{ENTITIES} + (\text{ENTITIES} / \text{SCOPE}))) + ((\text{POINTS AJUST.} + \text{POINTS AJUST.}) / \text{SCOPE})) + ((\text{POINTS NON AJUST.} - (\text{ENTITIES} * \text{TEAM EXP.})) / \text{MANAGER EXP.}))$	117579.30915826	63531056.2516439

Πίνακας A.13α: Αριθμητικά Desharnais A – Βάθος - Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Βάθος 5	86	0.562001	0.836068	6128168.541413	0.474256	1715.771903	0.354838	0.081957	0.722000	0.767888	5552945.003720	0.485280	1902.509693	0.2	0.452886
Βάθος 6	22	0.647263	0.763255	7598285.185541	0.574969	1813.284843	0.435483	-0.227404	0.704488	0.748025	7604692.904556	0.667662	2041.196288	0.466666	-0.116541
Βάθος 7	51	0.663442	0.777129	6712000.355162	0.561448	1753.778762	0.370967	-0.081157	0.719464	0.707224	13718799.705653	0.574835	2651.611708	0.333333	0.008909
Βάθος 7	55	0.620501	0.780327	7028712.748483	0.563084	1803.051569	0.387096	-0.193593	0.700959	0.712735	7677375.871958	0.563331	1875.361020	0.266666	-0.126540
Βάθος 7	65	0.638605	0.766923	8047042.287109	0.586703	1896.440470	0.483870	-0.259884	0.729216	0.702112	1933337.629131	0.689460	1108.060981	0.266666	-0.438887

Πίνακας A.13β: Αριθμητικά Desharnais A - Βάθος - Τιμές

Execution	A/A	Tree	Fitness	Total Fitness
Κόμβοι 20	55	$((((POINTS\ AJUST.) * POINTS\ AJUST.) + (TRANSACTIONS) * \log((DURATION\ MONTHS * (DURATION\ MONTHS * (DURATION\ MONTHS * DURATION\ MONTHS)))) * SCOPE)))$	110500.61240831	56307852.7139947
Κόμβοι 20	85	$((\log(SCOPE) + (DURATION\ MONTHS)) * (SCOPE + (POINTS\ NON\ AJUST. + (POINTS\ NON\ AJUST. + (((DURATION\ MONTHS) * \log(SCOPE)) - MANAGER\ EXP.))))))$	112370.59221026	57144846.1011433
Κόμβοι 20	100	$((((POINTS\ NON\ AJUST. + (((POINTS\ NON\ AJUST. + DURATION\ MONTHS) + DURATION\ MONTHS) + DURATION\ MONTHS) + DURATION\ MONTHS) / ((POINTS\ AJUST.)) * ((TRANSACTIONS))))$	110462.91871863	56280636.0967158
Κόμβοι 20	42	$((ENTITIES + (DURATION\ MONTHS * SCOPE)) * (((TRANSACTIONS * SCOPE)) + ((MANAGER\ EXP. * (DURATION\ MONTHS * DURATION\ MONTHS)) * MANAGER\ EXP.))))$	113897.53099085	59735410.4081626
Κόμβοι 28	31	$(((((SCOPE * (((SCOPE * DURATION\ MONTHS) - (DURATION\ MONTHS * \log(DURATION\ MONTHS))) + DURATION\ MONTHS))) + (SCOPE + (SCOPE + POINTS\ AJUST.))) * ((SCOPE * DURATION\ MONTHS)) - POINTS\ NON\ AJUST. - POINTS\ AJUST.))$	98341.28281491	51355041.1156327
Κόμβοι 28	78	$((SCOPE * ((\log(DURATION\ MONTHS) + ((SCOPE + TEAM\ EXP.)) * ((DURATION\ MONTHS * (SCOPE + ((DURATION\ MONTHS + (SCOPE + DURATION\ MONTHS)) + (SCOPE / \log(TEAM\ EXP.))))) + POINTS\ AJUST.)))$	114116.07624774	58432386.4433243
Κόμβοι 28	93	$(((((ENTITIES / MANAGER\ EXP.) / TEAM\ EXP.) / TEAM\ EXP.) + (POINTS\ AJUST. + (((\log(ENTITIES) / ((DURATION\ MONTHS) - TEAM\ EXP.) / MANAGER\ EXP.) + SCOPE) * DURATION\ MONTHS))) * \log((SCOPE * SCOPE)))$	99755.81670308	51424205.9165511
Κόμβοι 28	3	$((SCOPE + (DURATION\ MONTHS + POINTS\ AJUST.)) * \log(((DURATION\ MONTHS * DURATION\ MONTHS) * (DURATION\ MONTHS * ((DURATION\ MONTHS * (SCOPE + ENTITIES)) + ((SCOPE + ENTITIES) + ENTITIES))))))$	118675.94401957	60057296.7027905

Πίνακας Α.14α: Αριθμητικά Desharnais Α – Κόμβοι - Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Κόμβοι 20	55	0.649780	0.763618	7472767.254538	0.554878	1782.267942	0.435483	-0.203562	0.743403	0.729323	9916490.325055	0.636665	2151.114922	0.266666	0.014345
Κόμβοι 20	85	0.653451	0.764224	8006720.142972	0.546046	1812.428906	0.370967	-0.026241	0.693829	0.735068	6103480.870030	0.622811	2107.642818	0.333333	-0.248659
Κόμβοι 20	100	0.682441	0.733768	7864698.108743	0.617639	1781.659979	0.435483	-0.269674	0.745571	0.701294	11179106.285508	0.633466	2548.404969	0.266666	-0.075568
Κόμβοι 20	42	0.607916	0.795020	7308228.611696	0.546032	1837.056951	0.322580	-0.035830	0.939018	0.749407	7980914.112185	0.649181	2454.074481	0.066666	0.131096
Κόμβοι 28	31	0.589107	0.810563	5626059.902934	0.455544	1586.149722	0.467741	-0.036525	0.740722	0.773332	13460472.423403	0.663248	2477.961384	0.333333	-0.087695
Κόμβοι 28	78	0.631354	0.775382	7478043.789657	0.517565	1840.581874	0.403225	-0.039476	0.742501	0.710819	7342532.576667	0.572453	1867.426627	0.266666	0.168831
Κόμβοι 28	93	0.589837	0.806008	5862003.187022	0.476532	1608.964785	0.435483	-0.051317	0.756263	0.749284	12448554.691907	0.598135	2333.320040	0.266666	0.061805
Κόμβοι 28	3	0.682195	0.741197	9442534.217303	0.644675	1914.128129	0.451612	-0.360195	0.935547	0.735201	4849515.025211	0.678350	1733.856248	0.266666	-0.281667

Πίνακας A.146: Αριθμητικά Desharnais A - Κόμβοι - Τιμές

Βλέποντας τα αποτελέσματα τόσο των εκτελέσεων βάση βάθους αλλά και βάση κόμβων, το MRE έχει καλύτερη τιμή 0.485280 στην εκτέλεση 86 βάθους 5, το CC έχει καλύτερη τιμή 0.773332 στην εκτέλεση 31 των 28 κόμβων ενώ το NRMSE έχει τιμή 0.693829 στην εκτέλεση 85 των 20 κόμβων. Το BRE και εδώ έχει δώσει πολύ καλή τιμή 0.008909 στην εκτέλεση 51 του βάθους 7 ενώ PRED στην εκτέλεση 22 του βάθους 6 έδωσε μόλις 0.466666. Τα πιο πάνω αποτελέσματα εκτός από το γεγονός ότι είναι περιορισμένα σε αριθμό, όμως οι τιμές πρόβλεψης δεν πλησιάζουν καν αυτές του αρχείου Cocomo. Θέλοντας να βελτιώσουμε όσο το δυνατό περισσότερο την ακρίβεια της πρόβλεψης μας, η δεύτερη ομάδα εκτελέσεων στο ίδιο αρχείο θα γίνει με δεδομένα ελέγχου να είναι το 40% του αρχείου εισόδου και το υπόλοιπο 60% να είναι τα δεδομένα εκμάθησης όπως έχουμε κάνει και προηγουμένως (Desharnais B). Ας δούμε τα αποτελέσματα αυτά.

Execution	A/A	Tree	Fitness	Total Fitness
Βάθος 4	7	$(((\text{POINTS AJUST.} + \text{POINTS AJUST.}) + \text{SCOPE}) * ((\text{TRANSACTIONS} - \text{POINTS NON AJUST.})))$	76866.9761133	27360173.8188689
Βάθος 4	17	$(((\text{SCOPE}) * (\text{POINTS AJUST.} + \text{POINTS AJUST.})) + ((\text{POINTS NON AJUST.} + \text{TRANSACTIONS}) + (\text{POINTS NON AJUST.} + \text{POINTS NON AJUST.})))$	85938.92325081	30540049.7325218
Βάθος 4	53	$(((\text{SCOPE} + \text{SCOPE}) + (\text{POINTS NON AJUST.} + \text{DURATION MONTHS})) * ((\text{POINTS AJUST.} * \text{DURATION MONTHS})))$	87049.31193135	30884554.2637968
Βάθος 5	5	$(\text{POINTS NON AJUST.} * (((\text{DURATION MONTHS} * \text{DURATION MONTHS}) * (\text{DURATION MONTHS} * \text{SCOPE}))))$	101379.88578459	35692764.9455487
Βάθος 5	54	$(((((\text{DURATION MONTHS} * \text{SCOPE}) * (\text{DURATION MONTHS} * \text{SCOPE}))) * \text{POINTS AJUST.}))$	87389.08540222	31263358.517813

Πίνακας A.15α: Αριθμητικά Desharnais B – Βάθος 4 & 5 - Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Βάθος 4	7	0.628148	0.786609	5184732.154414	0.532719	1635.467576	0.382978	0.007645	0.702394	0.733343	12102666.917997	0.540657	2546.559948	0.233333	0.341629
Βάθος 4	17	0.660477	0.756897	7198507.092721	0.548130	1828.487728	0.255319	-0.014638	0.818298	0.704014	12768962.387437	0.569429	2396.317512	0.333333	0.226047
Βάθος 4	53	0.660774	0.755567	7391676.579389	0.529652	1852.113019	0.340425	-0.020231	0.697831	0.736166	9306559.338372	0.578314	2023.894943	0.366666	-0.008110
Βάθος 5	5	0.684264	0.749613	10217903.057852	0.530407	2157.018846	0.361702	0.064266	0.699383	0.738361	4908411.590061	0.631955	1594.154052	0.333333	-0.053725
Βάθος 5	54	0.674413	0.747417	9361239.201502	0.637515	1859.342242	0.425531	-0.281173	0.727159	0.700485	7041250.484553	0.649926	2041.404673	0.333333	-0.340302

Πίνακας A.15β: Αριθμητικά Desharnais B – Βάθος 4 & 5 - Τιμές

Execution	A/A	Tree	Fitness	Total Fitness
Βάθος 6	16	$(((((SCOPE * DURATION MONTHS) * SCOPE)) * (((POINTS NON AJUST. + DURATION MONTHS) + DURATION MONTHS) + ((POINTS AJUST. * MANAGER EXP.) * (SCOPE - DURATION MONTHS))))))$	83042.17675251	1.63720780225603 E+55
Βάθος 6	62	$((((DURATION MONTHS * (SCOPE + SCOPE))) * (((SCOPE * DURATION MONTHS) + (SCOPE + SCOPE)) + (TRANSACTIONS + (SCOPE + SCOPE)) + (((SCOPE + SCOPE) + SCOPE) + ((POINTS AJUST. + POINTS AJUST.) + (POINTS AJUST. / TEAM EXP.)))))$	66170.01154892	25601991.2934384
Βάθος 6	64	$(\log(((DURATION MONTHS * SCOPE) + SCOPE) * ((DURATION MONTHS - SCOPE) + ENTITIES))) * (((SCOPE * (DURATION MONTHS)) + (SCOPE * (DURATION MONTHS))) + POINTS NON AJUST.)$	72225.37835668	26282556.8097038
Βάθος 6	91	$(((((\log(DURATION MONTHS) + (POINTS AJUST.)) * (ENTITIES / (MANAGER EXP.)) + POINTS NON AJUST.) + ((DURATION MONTHS + \log(SCOPE)) * (POINTS AJUST. / (ENTITIES))) + SCOPE))$	74448.75370615	27683511.839963
Βάθος 6	92	$(DURATION MONTHS * ((SCOPE * \log((ENTITIES * POINTS NON AJUST.))) + ENTITIES))$	100334.80810508	35557631.5416298

Πίνακας A.16α: Αριθμητικά Desharnais B – Βάθος 6 - Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Βάθος 6	16	0.647403	0.761444	7673303.040661	0.490035	1766.854824	0.425531	-0.042592	0.716410	0.739671	8494101.379755	0.598844	2137.589307	0.366666	0.088146
Βάθος 6	62	0.641248	0.779405	5035595.842595	0.449120	1407.872586	0.425531	-0.043899	0.711890	0.733877	13355766.566507	0.652858	2532.895063	0.366666	0.036041
Βάθος 6	64	0.576607	0.813452	4880861.199322	0.564521	1536.710177	0.404255	-0.254573	0.701165	0.7086641	10869675.526721	0.627588	2413.075331	0.3	-0.235471
Βάθος 6	91	0.600666	0.799349	5119463.939763	0.598630	1584.016036	0.425531	-0.170998	0.804111	0.701312	15005922.528753	0.686154	2386.226143	0.366666	-0.292508
Βάθος 6	92	0.652962	0.790711	9673058.162801	0.547020	2134.783151	0.319148	0.034335	0.890040	0.733924	6877241.282941	0.675661	1932.011052	0.3	-0.013243

Πίνακας A.16β: Αριθμητικά Desharnais B – Βάθος 6 – Τιμές

Execution	A/A	Tree	Fitness	Total Fitness
Βάθος 7	2	$(POINTS\ AJUST.+((((SCOPE+TEAM\ EXP.)*DURATION\ MONTHS)*\log(TRANSACTIONS))+((MANAGER\ EXP.*MANAGER\ EXP.)*((MANAGER\ EXP.*MANAGER\ EXP.)+(MANAGER\ EXP.*TEAM\ EXP.))))+(POINTS\ AJUST.*\log(((ENTITIES+SCOPE)*(SCOPE))))))$	79717.51993312	29071390.7810879
Βάθος 7	7	$(((((DURATION\ MONTHS*SCOPE)/ENTITIES)*SCOPE)/MANAGER\ EXP.)+(SCOPE/DURATION\ MONTHS)+((SCOPE*DURATION\ MONTHS)+ENTITIES)+(SCOPE/ENTITIES)+ENTITIES))*((SCOPE+ENTITIES))$	71894.91403883	26385823.9771659
Βάθος 7	14	$((((ENTITIES)+(DURATION\ MONTHS*SCOPE)/(TEAM\ EXP.*SCOPE)))+((DURATION\ MONTHS*SCOPE)+(ENTITIES+(DURATION\ MONTHS*SCOPE))))+(DURATION\ MONTHS/((TEAM\ EXP./TRANSACTIONS)/TRANSACTIONS))*(((POINTS\ AJUST.)/(TEAM\ EXP.))/\log(\log(MANAGER\ EXP.))))$	72038.85862482	27155963.6993352
Βάθος 7	58	$((((POINTS\ NON\ AJUST.+SCOPE)+((ENTITIES+(SCOPE+DURATION\ MONTHS))/TEAM\ EXP.))+ENTITIES)*(((SCOPE+SCOPE)/TEAM\ EXP.-DURATION\ MONTHS)+(TRANSACTIONS*MANAGER\ EXP.)))$	72765.16025933	27101769.7023977
Βάθος 7	59	$((DURATION\ MONTHS*SCOPE)+((SCOPE)*((DURATION\ MONTHS*SCOPE)-((DURATION\ MONTHS-POINTS\ NON\ AJUST.)-POINTS\ NON\ AJUST.)))-(DURATION\ MONTHS*DURATION\ MONTHS))$	75675.93803664	7.73829229569474 E+89
Βάθος 7	74	$((((DURATION\ MONTHS*DURATION\ MONTHS)+(\log((TRANSACTIONS*TRANSACTIONS))*SCOPE))*(((DURATION\ MONTHS*ENTITIES)-(SCOPE+(POINTS\ AJUST.+POINTS\ AJUST.)))+(\log((DURATION\ MONTHS*ENTITIES))*ENTITIES))))$	85514.9451221	31229712.2841525

Πίνακας Α.17α: Αριθμητικά Desharnais Β – Βάθος 7 - Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Βάθος 7	2	0.635065	0.771562	7269197.644174	0.549602	1696.117445	0.425531	-0.085228	0.683258	0.743300	8042170.016987	0.611419	2140.646592	0.3	-0.110144
Βάθος 7	7	0.667694	0.753686	4830208.041690	0.482060	1529.679022	0.361702	-0.013943	0.670483	0.744526	12364380.713292	0.618463	2578.725405	0.266666	-0.179088
Βάθος 7	14	0.573908	0.820672	4852369.357627	0.412020	1532.741672	0.489361	-0.119544	0.633903	0.766944	8907867.439454	0.680964	2194.631019	0.066666	-0.050290
Βάθος 7	58	0.586110	0.805775	5983643.090955	0.551278	1548.194899	0.489361	-0.270403	0.766279	0.702868	10635977.367154	0.655499	2405.587930	0.233333	0.012827
Βάθος 7	59	0.647083	0.760779	5441947.021077	0.508737	1610.126341	0.425531	-0.036177	0.661477	0.755376	10914425.246052	0.577691	2254.795390	0.333333	0.069924
Βάθος 7	74	0.624783	0.781637	6575046.781313	0.449732	1819.466917	0.382978	-0.028470	0.748283	0.742923	10497696.715890	0.650323	1828.174945	0.366666	-0.253842

Πίνακας A.17β: Αριθμητικά Desharnais B – Βάθος 7 – Τιμές

Στην δεύτερη ομάδα εκτελέσεων του αρχείου Desharnais στις εκτελέσεις με βάση το βάθος του δένδρου, έχουμε το MRE να έχει καλύτερη τιμή 0.540657 στην εκτέλεση 7 βάθους 4, το CC 0.766944 και το NRMSE 0.633903 στην εκτέλεση 14 βάθους 7, το BRE -0.008110 στην εκτέλεση 53 βάθους 4 και τέλος το PRED 0.366666 σε διάφορες εκτελέσεις. Για το συντελεστή BRE βλέπουμε να έχουμε αρκετές αρνητικές τιμές. Το γεγονός αυτό οφείλεται στο ότι ο υπολογισμός της τιμής του δεν λαμβάνει υπόψη απόλυτη τιμή και επομένως όταν η υπολογιζόμενη τιμή είναι έστω και λίγο πιο μεγάλη από την πραγματική θα έχουμε αρνητική τιμή. Ας δούμε τι αποτελέσματα είχαμε και στις εκτελέσεις με βάση αριθμό κόμβων.

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Κόμβοι 16	73	0.512219	0.856771	4948159.851055	0.497742	1605.679355	0.446808	-0.066096	0.687162	0.726467	7571647.471433	0.572691	2089.332737	0.166666	-0.117047
Κόμβοι 20	4	0.606127	0.798976	6169960.256566	0.519869	1544.631472	0.425531	0.033823	0.683986	0.730693	8396077.825421	0.520538	2191.252538	0.333333	-0.034293
Κόμβοι 20	6	0.616780	0.795442	7110900.308754	0.523898	1731.154414	0.468085	-0.008711	0.732946	0.714727	8749391.305794	0.531510	2033.215448	0.233333	0.255680
Κόμβοι 20	40	0.634047	0.786650	7600617.705803	0.565575	1896.144634	0.361702	-0.192293	0.684846	0.751078	7221612.855445	0.670874	1886.852545	0.333333	-0.379088
Κόμβοι 20	41	0.617488	0.794793	5730519.122186	0.589942	1673.055566	0.425531	-0.194696	0.785405	0.709422	13589580.710733	0.675908	2579.054939	0.166666	-0.054839
Κόμβοι 20	47	0.611566	0.788110	7287292.292516	0.511147	1775.841049	0.425531	-0.064363	0.707433	0.748352	7478058.028953	0.555292	1953.747307	0.333333	0.104951
Κόμβοι 20	64	0.675522	0.740410	10172103.433217	0.582665	2193.243688	0.319148	-0.061197	0.678076	0.739627	4606103.704478	0.599387	1598.017809	0.4	-0.141886
Κόμβοι 20	66	0.591418	0.806671	4286711.355405	0.502789	1519.937217	0.468085	-0.168808	0.691429	0.716465	12638704.715405	0.656302	2436.514501	0.366666	-0.153016
Κόμβοι 20	76	0.618863	0.780570	5663898.597577	0.578017	1691.853621	0.404255	-0.226500	0.685032	0.729321	10470655.561626	0.678809	2379.605213	0.3	-0.270271
Κόμβοι 20	79	0.654199	0.750509	4973325.993088	0.583343	1482.673849	0.468085	-0.258863	0.699801	0.712482	12813873.081152	0.647379	2502.343532	0.3	-0.146884

Πίνακας Α.18β: Αριθμητικά Desharnais Β – Κόμβοι 16 & 20 - Τιμές

Execution	A/A	Tree	Fitness	Total Fitness
Κόμβοι 28	16	$(POINTS\ AJUST.*(((DURATION\ MONTHS*((TRANSACTIONS*DURATION\ MONTHS)+(TRANSACTIONS+(POINTS\ AJUST.*DURATION\ MONTHS))-(POINTS\ AJUST.*((TRANSACTIONS*(POINTS\ AJUST.*DURATION\ MONTHS)))))))+(TRANSACTIONS*DURATION\ MONTHS)))$	81644.60501001	29317492.2594047
Κόμβοι 28	22	$((((TRANSACTIONS+DURATION\ MONTHS)+DURATION\ MONTHS)+POINTS\ AJUST.*)*log(((SCOPE-((POINTS\ AJUST./log((DURATION\ MONTHS+ENTITIES)))-ENTITIES))*((SCOPE-((TRANSACTIONS/log(POINTS\ NON\ AJUST.))-ENTITIES))))))$	79658.78733903	29330199.5705396
Κόμβοι 28	39	$((((DURATION\ MONTHS+(DURATION\ MONTHS+TEAM\ EXP.))+(SCOPE+TEAM\ EXP.)+(SCOPE+((SCOPE+TEAM\ EXP.))+POINTS\ NON\ AJUST.)))*(((log(TEAM\ EXP.)+DURATION\ MONTHS)*DURATION\ MONTHS)*SCOPE))$	79167.66174329	28291091.9871513
Κόμβοι 28	42	$((((MANAGER\ EXP.*(ENTITIES*ENTITIES)))+(ENTITIES*DURATION\ MONTHS)+(log(((TRANSACTIONS-(log((MANAGER\ EXP.-(SCOPE*POINTS\ AJUST.)))*POINTS\ AJUST.)))+(TRANSACTIONS*POINTS\ AJUST.))*POINTS\ NON\ AJUST.))$	80736.23516323	28888922.9525318
Κόμβοι 28	43	$(SCOPE*SCOPE+((((DURATION\ MONTHS*(DURATION\ MONTHS*DURATION\ MONTHS))*DURATION\ MONTHS)+(TEAM\ EXP.-(TRANSACTIONS*SCOPE)))*POINTS\ AJUST.)$	91403.41288038	33215154.0304793
Κόμβοι 28	64	$(POINTS\ AJUST.*(((SCOPE)*(((ENTITIES+SCOPE)+SCOPE)/(log(log(log(log(SCOPE)))))))))))+(DURATION\ MONTHS)^(SCOPE))$	85070.54582914	32387278.5929903
Κόμβοι 28	67	$(((((log(ENTITIES)*DURATION\ MONTHS)+(ENTITIES*(log(log(POINTS\ NON\ AJUST.))*DURATION\ MONTHS))*log(SCOPE))-(POINTS\ AJUST.*DURATION\ MONTHS)))+(DURATION\ MONTHS))*POINTS\ AJUST.)$	67288.69129021	25944383.4020461
Κόμβοι 28	68	$(((((TEAM\ EXP.*(POINTS\ AJUST.^DURATION\ MONTHS))*MANAGER\ EXP.))+POINTS\ AJUST.))*((POINTS\ NON\ AJUST.+(((TRANSACTIONS+((TEAM\ EXP.*(POINTS\ AJUST.^DURATION\ MONTHS))*MANAGER\ EXP.))^SCOPE))))$	85747.70482721	30671685.7792758
Κόμβοι 28	76	$(log(POINTS\ AJUST.)*(SCOPE*DURATION\ MONTHS))+((TRANSACTIONS)*log((((DURATION\ MONTHS*MANAGER\ EXP.))-ENTITIES)-(log(SCOPE*DURATION\ MONTHS))*POINTS\ AJUST.)))*POINTS\ AJUST.)$	64856.62019636	24430987.9508515
Κόμβοι 28	95	$(TRANSACTIONS)*((log(TRANSACTIONS)*(DURATION\ MONTHS*(((TEAM\ EXP.^SCOPE)/POINTS\ NON\ AJUST.)/POINTS\ NON\ AJUST.)/(TEAM\ EXP.*(POINTS\ AJUST.)))))+ENTITIES)-ENTITIES)$	68539.0971338	26036541.7298101

Πίνακας Α.19α: Αριθμητικά Desharnais Β – Κόμβοι 28 - Παραγόμενες Εξισώσεις

Execution	A/A	TRAINING SET							TESTING SET						
		NRMSE	CC	MSE	MRE	MAE	PRED	BRE	NRMSE	CC	MSE	MRE	MAE	PRED	BRE
Κόμβοι 28	16	0.656669	0.769642	6134921.487998	0.532114	1737.119255	0.319148	-0.005042	0.736425	0.716346	12690861.180653	0.573109	2005.449149	0.466666	0.120896
Κόμβοι 28	22	0.565034	0.828748	6292322.308122	0.498339	1694.867815	0.468085	-0.083867	0.715337	0.708337	7475547.298377	0.621000	1984.146682	0.333333	-0.095417
Κόμβοι 28	39	0.639465	0.768851	6500054.586778	0.568243	1684.418334	0.340425	-0.182554	0.670617	0.732937	8811740.191119	0.597089	2096.136657	0.5	-0.283902
Κόμβοι 28	42	0.646303	0.788312	8436301.072836	0.542870	1717.792237	0.446808	-0.083826	0.719706	0.704195	7216352.362913	0.662307	2145.015859	0.233333	-0.081419
Κόμβοι 28	43	0.676548	0.749064	10215082.487245	0.614411	1944.753465	0.361702	-0.308458	0.723816	0.731530	4842437.120072	0.670106	1739.688118	0.233333	-0.357005
Κόμβοι 28	64	0.583913	0.810393	7022622.562866	0.530182	1810.011613	0.404255	-0.089633	0.673845	0.792431	5892469.341972	0.655668	1805.061782	0.333333	0.122802
Κόμβοι 28	67	0.545804	0.850706	3966909.277563	0.454480	1431.674282	0.382978	0.016277	0.710464	0.743956	12221447.698950	0.574409	2592.407485	0.266666	0.255714
Κόμβοι 28	68	0.678077	0.747164	9098117.421640	0.543373	1824.419251	0.446808	-0.122027	0.700461	0.712816	7055022.391507	0.633341	2030.603362	0.366666	-0.208627
Κόμβοι 28	76	0.501947	0.868362	4256923.312101	0.464326	1379.928089	0.468085	-0.132365	0.664827	0.749373	8315713.910656	0.652293	2266.175837	0.2	-0.101522
Κόμβοι 28	95	0.580535	0.824898	4832006.709327	0.523515	1458.278662	0.446808	-0.214631	0.736291	0.770196	12286148.09317	0.683637	2571.065737	0.333333	-0.229072

Πίνακας A.18β: Αριθμητικά Desharnais B – Κόμβοι 16 & 20 - Τιμές

Στις εκτελέσεις με βάση τον αριθμό κόμβων, έχουμε το MRE να έχει καλύτερη τιμή 0.520538 στην εκτέλεση 4 με 20 κόμβους, το CC 0.792431 στην εκτέλεση 64 με 28 κόμβους και το NRMSE 0.664827 στην εκτέλεση 76 με 28 κόμβους. Το BRE και πάλι με αρνητική τιμή -0.034293 στην εκτέλεση 4 με 20 κόμβους και τέλος το PRED 0.466666 στην εκτέλεση 16 με 28 κόμβους.

A.4 Αποτελέσματα Λογικών Εκτελέσεων – Αρχείο Cocomo

Execution	A/A	Tree	TRAINING SET			TESTING SET	
			Hit Ratio	Mean	Std	Hit Ratio	Success Ratio
Μέγεθος 2	2	((LOC<=1150) (SCED<=1.23))	51/51	733.29803922	1993.51644272	12/12	12/12
Μέγεθος 2	3	((LEXP<=1.14) &&(LOC<=1150))	51/51	789.21960784	2002.19796244	12/12	12/12
Μέγεθος 2	4	IF ((DATA<=1.16) THEN ((AEXP<=1.29)) ELSE ((TOOL<=1.24)))	51/51	703.35686275	2006.56910873	12/12	12/12
Μέγεθος 2	7	IF ((PCAP>0.7) THEN ((TURN<=1.15)) ELSE ((VIRT<=1.3)))	51/51	755.45490196	1991.38659002	12/12	12/12
Μέγεθος 2	8	((VEXP>0.9) (ACAP<=1.46))	51/51	780.68431373	2007.08310704	12/12	12/12
Μέγεθος 2	13	IF ((PCAP<=1.42) THEN ((TIME<=1.66)) ELSE ((RELY>0.75)))	51/51	799.72941176	2007.0675393	12/12	12/12
Μέγεθος 2	15	IF ((CPLX<=1.65) THEN ((VIRT<=1.3)) ELSE ((TURN>0.87)))	51/51	804.84705882	2006.05073668	12/12	12/12
Μέγεθος 2	17	((TIME<=1.66) (STOR<=1.56))	51/51	813.10196078	2004.45255554	12/12	12/12
Μέγεθος 2	24	IF ((TOOL<=1.24) THEN ((MODP<=1.24)) ELSE ((AEXP>0.82)))	51/51	774.41568627	2006.09104324	12/12	12/12
Μέγεθος 2	28	((AEXP<=1.29) &&(RELY<=1.4))	51/51	745.96470588	1989.45608967	12/12	12/12
Μέγεθος 2	29	IF ((TOOL<=1.24) THEN ((ACAP<=1.46)) ELSE ((RELY<=1.4)))	51/51	749.82941176	1995.04444966	12/12	12/12
Μέγεθος 2	30	IF ((DATA<=1.16) THEN ((ACAP<=1.46)) ELSE ((LOC<=1150)))	51/51	787.27843137	2011.77301079	12/12	12/12
Μέγεθος 2	33	((LOC<=1150) (STOR>1))	51/51	767.86862745	2013.06633577	12/12	12/12
Μέγεθος 2	34	((TIME<=1.66) &&(SCED<=1.23))	51/51	745.66470588	1996.73260742	12/12	12/12
Μέγεθος 2	35	IF ((CPLX<=1.65) THEN ((TIME<=1.66)) ELSE ((TURN>0.87)))	51/51	777.29215686	2006.91430239	12/12	12/12
Μέγεθος 2	38	((TIME<=1.66) (RELY>0.75))	51/51	801.39607843	2005.01595355	12/12	12/12
Μέγεθος 2	41	((VIRT<=1.3) (AEXP>0.82))	51/51	804.12156863	2004.4037854	12/12	12/12

Μέγεθος 2	42	IF ((STOR<=1.56) THEN ((SCED<=1.23)) ELSE ((LOC<=1150)))	51/51	785.10196078	2011.37800311	12/12	12/12
Μέγεθος 2	43	((ACAP<=1.46) &&(STOR<=1.56))	51/51	797.27843137	2008.64664063	12/12	12/12
Μέγεθος 2	45	((STOR<=1.56) (LOC>1.98))	51/51	766.35098039	2006.29171223	12/12	12/12
Μέγεθος 2	46	((CPLX<=1.65) &&(SCED<=1.23))	51/51	810.74901961	2004.8591097	12/12	12/12
Μέγεθος 2	47	((ACAP<=1.46) (DATA<=1.16))	51/51	806.39607843	2007.24391492	12/12	12/12
Μέγεθος 2	49	((RELY<=1.4) (DATA<=1.16))	51/51	756.84705882	2009.2674342	12/12	12/12
Μέγεθος 2	50	IF ((STOR<=1.56) THEN ((MODP<=1.24)) ELSE ((AEXP<=1.29)))	51/51	760.43529412	1994.49166173	12/12	12/12
Μέγεθος 2	1	((DATA<=1.16) (MODP>0.82))	51/51	789.67058824	2007.76633384	12/12	12/12
Μέγεθος 2	8	IF ((PCAP<=1.42) THEN ((VEXP<=1.21)) ELSE ((PCAP>0.7)))	51/51	784.79019608	1996.87333622	12/12	12/12
Μέγεθος 2	12	IF ((AEXP>0.82) THEN ((VEXP<=1.21)) ELSE ((LEXP<=1.14)))	51/51	783.86666667	2008.27902241	12/12	12/12
Μέγεθος 2	15	((MODP<=1.24) (STOR>1))	51/51	785.65098039	2006.74288581	12/12	12/12

Πίνακας Α.19: Λογικά Cosoto - Μέγεθος Δένδρου 2

Execution	A/A	Tree	TRAINING SET			TESTING SET	
			Hit Ratio	Mean	Std	Hit Ratio	Success Ratio
Μέγεθος 3	4	(IF ((DATA>0.94) THEN ((STOR<=1.56) ELSE ((TOOL>0.83))) ((AEXP<=1.29) (CPLX<=1.65)))	51/51	792.66666667	2001.97292356	12/12	12/12
Μέγεθος 3	6	(IF ((LEXP>0.95) THEN ((STOR<=1.56) ELSE ((VIRT<=1.3))) ((TIME<=1.66) (LEXP>0.95)))	51/51	797.67058824	2006.85822801	12/12	12/12
Μέγεθος 3	9	((((STOR<=1.56) NAND(VEXP>0.9)) NAND((TIME<=1.66) XOR(ACAP<=1.46)))	51/51	788.51568627	2008.69636296	12/12	12/12
Μέγεθος 3	11	(IF ((VIRT>0.87) THEN ((MODP<=1.24) ELSE ((TURN>0.87))) NAND((PCAP<=1.42)))	51/51	798.37647059	2008.85915719	12/12	12/12
Μέγεθος 3	12	((((LOC>1.98) NOR(SCED<=1.23)) NAND((ACAP<=1.46)))	51/51	766.29215686	1992.60293414	12/12	12/12
Μέγεθος 3	18	((((RELY>0.75) NOR(VIRT<=1.3)))	51/51	737.10196078	1990.86471047	12/12	12/12
Μέγεθος 3	20	((((DATA>0.94) NAND((LOC>1.98) NOR(SCED<=1.23)))	51/51	717.58627451	1993.36536882	12/12	12/12
Μέγεθος 3	22	((((ACAP<=1.46) NOR(DATA<=1.16)))	51/51	746.23921569	2011.72675968	12/12	12/12
Μέγεθος 3	25	(IF ((VEXP<=1.21) THEN ((LOC<=1150) ELSE ((TOOL>0.83))) XOR((VIRT<=1.3) NOR(AEXP<=1.29)))	51/51	737.46862745	2004.0559324	12/12	12/12

Πίνακας Α.20: Λογικά Cosmo - Μέγεθος Δένδρου 3

Execution	A/A	Tree	TRAINING SET			TESTING SET	
			Hit Ratio	Mean	Std	Hit Ratio	Success Ratio
Μέγεθος 4	3	(IF (((VEXP<=1.21)) THEN (((LEXP<=1.14) NOR(TURN>0.87))) ELSE (((AEXP>0.82) NOR(TIME<=1.66)))) NAND(IF ((ACAP>0.71) THEN ((LOC>1.98) ELSE ((VEXP>0.9))) ((LOC<=1150) NAND(VIRT>0.87))))	51/51	740.05686275	1998.00552374	12/12	12/12
Μέγεθος 4	4	IF (((TIME>1) NOR(AEXP>0.82)) XOR((TURN<=1.15) (TURN>0.87))) THEN (((CPLX<=1.65) NAND((STOR<=1.56) (LEXP<=1.14)))) ELSE ((IF ((SCED<=1.23) THEN ((LOC>1.98) ELSE ((SCED>1))) ((PCAP<=1.42) &&(LOC<=1150))))	51/51	820.09607843	2001.98555109	12/12	12/12
Μέγεθος 4	7	((((ACAP<=1.46) NOR(TIME<=1.66)) ((LEXP<=1.14) XOR(SCED<=1.23))))	51/51	784.21960784	2007.14052343	12/12	12/12
Μέγεθος 4	8	((((STOR<=1.56) &&(DATA>0.94)) &&((SCED<=1.23) XOR(TURN<=1.15))) XOR(((CPLX>0.7) NAND((ACAP>0.71) NOR(RELY<=1.4))))	51/51	795.29803922	2006.77948749	12/12	12/12
Μέγεθος 4	9	((((SCED<=1.23)) &&((RELY<=1.4) XOR(LEXP<=1.14))))	51/51	771.2	2009.85916253	12/12	12/12
Μέγεθος 4	10	(((((MODP<=1.24) XOR(VIRT<=1.3)) &&((STOR>1) NAND(VIRT>0.87))) XOR(((SCED<=1.23) NOR(MODP<=1.24))))	51/51	777.27254902	2008.30830512	12/12	12/12
Μέγεθος 4	12	(((((AEXP>0.82) NOR(STOR>1)) ((CPLX<=1.65) XOR(LOC<=1150))) (((SCED<=1.23) NAND((VIRT>0.87) (RELY>0.75))))	51/51	825.35882353	2001.23988929	12/12	12/12
Μέγεθος 4	13	(((((MODP<=1.24) XOR(PCAP<=1.42)) XOR((STOR>1) XOR(SCED>1))) NAND(((DATA<=1.16) NOR(SCED<=1.23)) &&(LOC>1.98) NAND(VIRT>0.87))))	51/51	775.84705882	2002.92930233	12/12	12/12
Μέγεθος 4	14	(((((ACAP>0.71) NAND((MODP<=1.24))) (((RELY>0.75) &&(TURN<=1.15))))	51/51	782.70980392	1997.4479648	12/12	12/12
Μέγεθος 4	21	(((((DATA>0.94)) NAND((LOC>1.98) NOR(STOR>1))) (((TIME<=1.66) NAND(VIRT>0.87)) &&(CPLX<=1.65) NOR(PCAP>0.7))))	51/51	809.70980392	2005.56439839	12/12	12/12
Μέγεθος 4	24	(((((TIME<=1.66))) IF (((PCAP>0.7) XOR(TIME>1)) THEN (IF ((LEXP>0.95) THEN ((ACAP<=1.46) ELSE ((STOR>1)))) ELSE (((STOR>1) &&(STOR<=1.56))))	51/51	825.74901961	2001.26134469	12/12	12/12
Μέγεθος 4	25	(((((ACAP<=1.46) NAND(AEXP<=1.29))) (((TURN<=1.15) &&(LOC<=1150)) NOR((AEXP<=1.29) XOR(DATA<=1.16))))	51/51	731.88627451	1998.07581258	12/12	12/12

Πίνακας Α.21: Λογικά Cosmo - Μέγεθος Δένδρου 4

Execution	A/A	Tree	TRAINING SET			TESTING SET	
			Hit Ratio	Mean	Std	Hit Ratio	Success Ratio
Μέγεθος 5	1	((IF (((LOC<=1150) NOR(TOOL<=1.24)) THEN (IF ((VEXP<=1.21) THEN ((TURN>0.87)) ELSE ((ACAP<=1.46)))) ELSE (((ACAP>0.71) XOR(TIME<=1.66)))))) (((LEXP<=1.14)) NOR((SCED<=1.23))) XOR(((RELY<=1.4) ((TURN<=1.15)) NOR((SCED>1) XOR(DATA>0.94))))))	51/51	794.35686275	2009.22021195	12/12	12/12
Μέγεθος 5	2	(((((LOC>1.98) &&(ACAP<=1.46)) NAND((AEXP<=1.29) NAND(SCED<=1.23))) &&(IF ((VIRT<=1.3) THEN ((TOOL<=1.24)) ELSE ((VIRT<=1.3))) XOR((MODP<=1.24) NOR(AEXP<=1.29)))) NANDIF (((CPLX<=1.65) &&(RELY<=1.4)) &&(IF ((TURN>0.87) THEN ((SCED>1) ELSE ((TOOL<=1.24)))) THEN (((LEXP<=1.14) NOR(PCAP>0.7)) &&(IF ((AEXP<=1.29) THEN ((VIRT<=1.3) ELSE ((LOC<=1150)))) ELSE (((CPLX<=1.65)) XOR((VEXP<=1.21))))))	51/51	720.51372549	1998.24016395	12/12	12/12
Μέγεθος 5	10	IF (((((TOOL<=1.24) ((PCAP<=1.42)) &&(MODP<=1.24) NOR(RELY>0.75))) &&(((DATA<=1.16) XOR(ACAP>0.71)) ((LOC>1.98) XOR(LEXP<=1.14)))) THEN (((PCAP<=1.42) XOR(TURN<=1.15)) NAND((TURN<=1.15) NOR(ACAP<=1.46)))) ELSE (((LEXP<=1.14)) &&(TOOL<=1.24) ((SCED<=1.23))) NOR(((TIME<=1.66) (ACAP<=1.46))))))	51/51	761.94509804	2006.46579411	12/12	12/12
Μέγεθος 5	15	((((IF ((VIRT<=1.3) THEN ((VIRT<=1.3)) ELSE ((LOC>1.98))) XOR((STOR<=1.56) XOR(LEXP<=1.14))))))	51/51	813.23921569	2005.25945943	12/12	12/12
Μέγεθος 5	18	((IF (((PCAP>0.7) NOR(TIME>1)) THEN (((RELY<=1.4) NAND(RELY<=1.4))) ELSE (((CPLX<=1.65) &&(CPLX>0.7)))) NOR(((SCED<=1.23) NAND((LEXP<=1.14))))))	51/51	721.11568627	1987.47129955	12/12	12/12
Μέγεθος 5	19	(((((PCAP>0.7) ((STOR>1)) XOR((ACAP>0.71) NAND(STOR>1))) NOR(((SCED<=1.23) XOR(SCED>1) NAND((CPLX>0.7)))) NANDIF (((VEXP>0.9) &&(VIRT>0.87)) NOR((DATA<=1.16))) THEN ((IF ((VEXP<=1.21) THEN ((CPLX<=1.65)) ELSE ((DATA>0.94))) NAND((VIRT<=1.3) NAND(STOR>1))) ELSE (((CPLX>0.7) XOR(VIRT>0.87)) NOR((VIRT<=1.3) &&(TOOL>0.83))))))	51/51	709.94509804	1982.53385316	12/12	12/12
Μέγεθος 5	21	((((IF ((RELY<=1.4) THEN ((CPLX<=1.65)) ELSE ((STOR<=1.56))) &&(VIRT<=1.3))) NOR(IF ((MODP<=1.24) THEN ((TURN>0.87)) ELSE ((VEXP<=1.21))) NAND((LOC<=1150) NAND(DATA<=1.16))) NANDIF (((VEXP>0.9) &&(STOR<=1.56))) THEN (((VEXP>0.9) XOR(STOR>1))) ELSE (((TOOL<=1.24) NAND(AEXP>0.82)) ((DATA>0.94) &&(TURN<=1.15))))))	51/51	780.0627451	2004.9449364	12/12	12/12
Μέγεθος 5	22	(((((LOC<=1150)) ((PCAP>0.7) (ACAP>0.71))) XOR(((CPLX>0.7) ((TOOL>0.83)) NOR((PCAP>0.7) NAND(STOR>1)))) NAND(((VEXP>0.9) &&(AEXP<=1.29)) NOR((PCAP>0.7) NOR(RELY<=1.4))) &&(IF ((DATA<=1.16) THEN ((MODP<=1.24)) ELSE ((PCAP>0.7))) NOR((MODP>0.82) NAND(ACAP>0.71))))))	51/51	755.14117647	2005.337573	12/12	12/12
Μέγεθος 5	23	((IF (((AEXP>0.82) &&(MODP<=1.24)) &&(PCAP>0.7) XOR(TIME>1))) THEN ((IF ((TOOL>0.83) THEN ((LEXP>0.95)) ELSE ((PCAP>0.7)))) ELSE ((IF ((RELY<=1.4) THEN ((TOOL<=1.24)) ELSE ((DATA>0.94)))) (((TOOL<=1.24) ((SCED>1)) NAND((LEXP>0.95) NOR(PCAP<=1.42))) NAND(((TURN>0.87) ((ACAP<=1.46)) &&(DATA<=1.16) NOR(LEXP<=1.14))))))	51/51	791.25882353	2007.27037682	12/12	12/12
Μέγεθος 5	25	(((((TIME<=1.66)) XORIF ((TURN<=1.15) THEN ((TURN<=1.15)) ELSE ((LEXP<=1.14)))) XOR(((CPLX>0.7)) NOR(TIME>1) ((PCAP<=1.42)))) (((DATA<=1.16)) ((IF ((AEXP>0.82) THEN ((AEXP>0.82)) ELSE ((LEXP<=1.14)))) XOR(((PCAP<=1.42) NOR(VEXP>0.9)) XOR((PCAP<=1.42) NAND(AEXP>0.82))))))	51/51	782.72352941	2009.41133426	12/12	12/12

Πίνακας Α.22: Λογικά Cosoto - Μέγεθος Δένδρου 5

A.5 Αποτελέσματα Λογικών Εκτελέσεων – Αρχείο Desharnais

Execution	A/A	Tree	TRAINING SET			TESTING SET	
			Hit Ratio	Mean	Std	Hit Ratio	Success Ratio
Μέγεθος 2	5	IF ((POINTS NON AJUST.<=1116) THEN ((POINTS AJUST.<=1127)) ELSE ((TRANSACTIONS<=886)))	62/62	5002.06451613	4439.70880699	15/15	14/15
Μέγεθος 2	8	((POINTS AJUST.>73) (MANAGER EXP.<=7))	62/62	4790.37096774	4358.52621442	15/15	14/15
Μέγεθος 2	11	((SCOPE<=52) (DURATION MONTHS>1))	62/62	5052.91935484	4469.11134416	15/15	14/15
Μέγεθος 2	17	((POINTS AJUST.<=1127) (TRANSACTIONS>9))	62/62	4927.66129032	4394.50970034	15/15	14/15
Μέγεθος 2	21	((DURATION MONTHS<=36) (TEAM EXP.<=4))	62/62	4884.30645161	4350.25723599	15/15	14/15
Μέγεθος 2	1	IF ((POINTS AJUST.<=1127) THEN ((ENTITIES<=387)) ELSE ((TRANSACTIONS<=886)))	62/62	4929.91935484	4492.84611792	15/15	15/15
Μέγεθος 2	4	((ENTITIES>7) (POINTS AJUST.<=1127))	62/62	5105.48387097	4485.71886427	15/15	14/15
Μέγεθος 2	6	IF ((ENTITIES>7) THEN ((SCOPE<=52)) ELSE ((TEAM EXP.<=4)))	62/62	4975.75806452	4558.0263621	15/15	15/15

Πίνακας A.23: Λογικά Desharnais - Μέγεθος Δένδρου 2

Execution	A/A	Tree	TRAINING SET			TESTING SET	
			Hit Ratio	Mean	Std	Hit Ratio	Success Ratio
Μέγεθος 3	12	((((POINTS AJUST.<=1127)) NAND((ENTITIES<=387)))	62/62	4855.06451613	4311.10417894	15/15	14/15
Μέγεθος 3	15	((((SCOPE<=52) XOR(POINTS AJUST.<=1127)) NAND((TEAM EXP.<=4)))	62/62	4988.17741935	4368.13951289	15/15	14/15
Μέγεθος 3	21	(IF ((TEAM EXP.>0) THEN ((SCOPE>5)) ELSE ((DURATION MONTHS<=36))) ((MANAGER EXP.>0) (ENTITIES>7)))	62/62	5008.9516129	4458.85089007	15/15	14/15
Μέγεθος 3	22	((((TRANSACTIONS<=886)))	62/62	5256.20967742	4524.16138823	15/15	14/15
Μέγεθος 3	9	((((ENTITIES<=387) NOR(TEAM EXP.<=4)) NAND((POINTS AJUST.<=1127) NAND(TEAM EXP.>0)))	62/62	4981.96774194	4556.28975089	15/15	14/15

Πίνακας A.24: Λογικά Desharnais - Μέγεθος Δένδρου 3

Execution	A/A	Tree	TRAINING SET			TESTING SET	
			Hit Ratio	Mean	Std	Hit Ratio	Success Ratio
Μέγεθος 4	3	(((SCOPE>5) &&(ENTITIES<=387)) NOR(SCOPE>5)) (((SCOPE<=52)) NOR((TEAM EXP.<=4)))	62/62	4757.90322581	4516.25948052	15/15	14/15
Μέγεθος 4	6	(((MANAGER EXP.<=7)) XOR((TRANSACTIONS>9) (POINTS AJUST.<=1127))) XOR(((TEAM EXP.<=4) NAND(POINTS NON AJUST.>62)) NOR((POINTS AJUST.<=1127) (TEAM EXP.>0))))	62/62	5068.17741935	4527.53360109	15/15	15/15
Μέγεθος 4	7	(((DURATION MONTHS>1) NAND(TRANSACTIONS>9)) NAND((MANAGER EXP.<=7) NOR(DURATION MONTHS>1))) &&(((DURATION MONTHS<=36) XOR(SCOPE>5)) NAND((POINTS AJUST.<=1127) NOR(POINTS AJUST.<=1127))))	62/62	5040.17741935	4483.70491024	15/15	14/15
Μέγεθος 4	8	(((DURATION MONTHS<=36) NOR(DURATION MONTHS>1))) &&(((POINTS NON AJUST.>62) (MANAGER EXP.<=7)) (POINTS NON AJUST.<=1116) XOR(TRANSACTIONS>9))))	62/62	5139.69354839	4444.61147904	15/15	15/15
Μέγεθος 4	9	(((POINTS AJUST.>73) NAND(ENTITIES>7)) NAND((TRANSACTIONS>9) NOR(SCOPE<=52))) (((TRANSACTIONS>9) NAND(TEAM EXP.<=4)) NAND((ENTITIES>7) (DURATION MONTHS>1))))	62/62	5065.74193548	4390.14140306	15/15	14/15
Μέγεθος 4	21	(((MANAGER EXP.>0) &&(SCOPE<=52)) NORIF ((POINTS AJUST.>73) THEN ((MANAGER EXP.>0)) ELSE ((DURATION MONTHS<=36)))) (((POINTS NON AJUST.>62) (TEAM EXP.>0)) &&((ENTITIES<=387) &&(POINTS AJUST.<=1127))))	62/62	4966.72580645	4434.07126383	15/15	14/15
Μέγεθος 4	26	((IF ((DURATION MONTHS<=36) THEN ((MANAGER EXP.>0)) ELSE ((DURATION MONTHS>1))) XOR((ENTITIES>7) (ENTITIES>7))) NAND(((TEAM EXP.<=4) NAND(SCOPE>5)) &&IF ((ENTITIES<=387) THEN ((MANAGER EXP.>0)) ELSE ((DURATION MONTHS<=36))))))	62/62	4917.61290323	4321.90047129	15/15	14/15
Μέγεθος 4	27	(((ENTITIES>7) &&(POINTS NON AJUST.>62))) NAND(((MANAGER EXP.<=7) NOR(ENTITIES<=387)) &&((ENTITIES<=387))))	62/62	5172.77419355	4544.14185453	15/15	15/15
Μέγεθος 4	28	(((POINTS NON AJUST.>62)) NOR((MANAGER EXP.>0) (TRANSACTIONS<=886))) NAND(((DURATION MONTHS<=36) (SCOPE<=52)) XOR((MANAGER EXP.>0) (ENTITIES>7))))	62/62	5074.66129032	4484.16912253	15/15	15/15
Μέγεθος 4	29	IF (IF (((TRANSACTIONS<=886) XOR(DURATION MONTHS<=36)) THEN (((TRANSACTIONS<=886) NOR(POINTS AJUST.>73))) ELSE (((MANAGER EXP.>0) NOR(POINTS AJUST.>73)))) THEN (((DURATION MONTHS>1) NOR(DURATION MONTHS>1)) &&IF ((POINTS NON AJUST.>62) THEN ((MANAGER EXP.>0)) ELSE ((DURATION MONTHS>1)))) ELSE (((TEAM EXP.>0) &&(TRANSACTIONS>9)) NAND((MANAGER EXP.<=7) XOR(ENTITIES<=387))))	62/62	5092.83870968	4438.62003808	15/15	14/15

Πίνακας Α.25: Λογικά Desharnais - Μέγεθος Δένδρου 4

Execution	A/A	Tree	TRAINING SET			TESTING SET	
			Hit Ratio	Mean	Std	Hit Ratio	Success Ratio
Μέγεθος 5	1	((IF (((DURATION MONTHS<=36) XOR(DURATION MONTHS<=36)) THEN (((POINTS NON AJUST.<=1116))) ELSE (((TEAM EXP.<=4) NAND(TRANSACTIONS<=886)))) &&(((TEAM EXP.>0) NOR(POINTS NON AJUST.<=1116)) XOR((TRANSACTIONS<=886))) NAND((((POINTS NON AJUST.<=1116) XOR(TEAM EXP.<=4)) NOR((POINTS AJUST.<=1127) XOR(TEAM EXP.>0))) XOR(IF ((POINTS NON AJUST.<=1116) THEN ((DURATION MONTHS<=36)) ELSE ((DURATION MONTHS>1))) NAND((MANAGER EXP.>0) XOR(TEAM EXP.<=4))))))	62/62	4941.20967742	4526.95875959	15/15	15/15
Μέγεθος 5	5	(((((DURATION MONTHS<=36) NAND(ENTITIES>7)) XOR((SCOPE>5) &&(TEAM EXP.<=4)))) NAND((((MANAGER EXP.<=7) XOR(POINTS AJUST.<=1127)) NAND((POINTS NON AJUST.>62) XOR(SCOPE>5))))))	62/62	4944.37096774	4406.49881549	15/15	14/15
Μέγεθος 5	9	(((((SCOPE<=52)) XORIF ((TRANSACTIONS<=886) THEN ((TRANSACTIONS<=886)) ELSE ((ENTITIES>7)))) (((TEAM EXP.<=4) XOR(ENTITIES<=387)) NAND((POINTS AJUST.>73) &&(ENTITIES<=387)))) &&((((TRANSACTIONS<=886) XOR(TRANSACTIONS<=886)) NAND((POINTS AJUST.>73))) (((SCOPE>5) (DURATION MONTHS>1)) NOR((SCOPE<=52) NOR(TEAM EXP.>0))))))	62/62	5031.91935484	4485.11555967	15/15	14/15
Μέγεθος 5	10	((((IF ((ENTITIES<=387) THEN ((POINTS AJUST.<=1127)) ELSE ((POINTS AJUST.<=1127))) &&(TEAM EXP.<=4) &&(ENTITIES>7))) NOR(((POINTS NON AJUST.<=1116) NOR(POINTS NON AJUST.>62)) NANDIF ((SCOPE>5) THEN ((TRANSACTIONS>9)) ELSE ((ENTITIES<=387)))))) NAND(IF (((POINTS AJUST.>73) &&(TRANSACTIONS>9)) THEN ((SCOPE>5)) ELSE (((MANAGER EXP.<=7) NAND(POINTS NON AJUST.>62)))) NOR(((MANAGER EXP.>0) XOR(POINTS NON AJUST.<=1116)) NOR((ENTITIES>7) (MANAGER EXP.>0))))))	62/62	4976.82258065	4326.73184704	15/15	14/15
Μέγεθος 5	12	(((((TEAM EXP.<=4) &&(DURATION MONTHS<=36))) XOR(IF ((SCOPE>5) THEN ((TEAM EXP.<=4)) ELSE ((POINTS NON AJUST.<=1116))) NAND((TEAM EXP.<=4)))) (IF (((ENTITIES>7) (TRANSACTIONS<=886)) THEN (IF ((POINTS AJUST.>73) THEN ((TRANSACTIONS<=886)) ELSE ((TRANSACTIONS>9)))) ELSE (((POINTS AJUST.<=1127) NAND(TEAM EXP.<=4)))) NAND((((SCOPE>5) (MANAGER EXP.>0))))))	62/62	4815.5483871	4458.62537692	15/15	14/15
Μέγεθος 5	14	((((((DURATION MONTHS>1) NOR(TRANSACTIONS<=886)) NAND((TRANSACTIONS<=886) (POINTS NON AJUST.>62))) (((DURATION MONTHS<=36) XOR(TEAM EXP.>0)) ((DURATION MONTHS<=36)))) (((TEAM EXP.>0) (POINTS NON AJUST.<=1116)) &&((POINTS AJUST.<=1127) &&(TEAM EXP.>0))) &&(((SCOPE<=52) &&(ENTITIES<=387))))))	62/62	5036.16129032	4535.74566752	15/15	15/15
Μέγεθος 5	18	(IF (((((DURATION MONTHS<=36) NANDIF ((DURATION MONTHS>1) THEN ((TRANSACTIONS<=886)) ELSE ((TRANSACTIONS>9)))) &&(IF ((POINTS NON AJUST.<=1116) THEN ((SCOPE<=52)) ELSE ((TRANSACTIONS<=886)))))) THEN (((((MANAGER EXP.>0) (IF ((TRANSACTIONS>9) THEN ((POINTS AJUST.>73)) ELSE ((POINTS AJUST.<=1127)))) XOR(IF ((DURATION MONTHS<=36) THEN ((POINTS NON AJUST.<=1116)) ELSE ((POINTS NON AJUST.>62))) ((DURATION	62/62	4905.41935484	4470.42781738	15/15	14/15

		MONTHS<=36) NAND(DURATION MONTHS<=36)))) ELSE (((((MANAGER EXP.>0) (POINTS AJUST.>73)) NOR((TEAM EXP.>0) (POINTS NON AJUST.<=1116))) (((POINTS AJUST.>73) (SCOPE>5) ((ENTITIES<=387) NAND(DURATION MONTHS>1))))))					
Μέγεθος 5	19	IF ((((((POINTS NON AJUST.<=1116) (ENTITIES<=387)) &&((DURATION MONTHS<=36) &&(POINTS NON AJUST.<=1116))) XOR(((SCOPE>5) NOR(MANAGER EXP.>0)))) THEN (((((POINTS NON AJUST.>62) &&(TEAM EXP.>0)) &&((ENTITIES<=387) &&(TEAM EXP.<=4))) &&(((DURATION MONTHS>1) XOR(POINTS AJUST.>73)) NORIF ((MANAGER EXP.>0) THEN ((MANAGER EXP.<=7) ELSE ((ENTITIES<=387)))))) ELSE (IF (((DURATION MONTHS<=36) &&(SCOPE>5)) NOR((POINTS AJUST.<=1127) (DURATION MONTHS<=36))) THEN (((TEAM EXP.>0) (POINTS AJUST.>73)) &&((ENTITIES>7) NOR(POINTS AJUST.<=1127)))) ELSE (((MANAGER EXP.<=7))))))	62/62	4998.22580645	4484.48502597	15/15	15/15
Μέγεθος 5	23	(((SCOPE>5) &&(POINTS NON AJUST.>62)) ((TEAM EXP.>0) NAND(TRANSACTIONS>9))) NAND(((MANAGER EXP.>0) NAND(ENTITIES<=387)) &&IF ((TRANSACTIONS<=886) THEN ((SCOPE<=52) ELSE ((ENTITIES<=387)))) IF (((TEAM EXP.<=4) NAND(SCOPE<=52))) THEN (((TRANSACTIONS>9) (DURATION MONTHS>1) XOR((POINTS NON AJUST.>62) NOR(TEAM EXP.>0)))) ELSE (((POINTS AJUST.<=1127) (MANAGER EXP.<=7) NAND((POINTS AJUST.>73) (SCOPE>5))))))	62/62	4997.83870968	4397.4667603	15/15	14/15
Μέγεθος 5	24	(((POINTS AJUST.<=1127) NAND(SCOPE<=52)) ((ENTITIES>7) NOR(POINTS AJUST.<=1127))) (IF ((MANAGER EXP.<=7) THEN ((TRANSACTIONS>9) ELSE ((TEAM EXP.<=4))) ((TEAM EXP.<=4) NAND(TRANSACTIONS<=886)))) NAND(((DURATION MONTHS<=36) &&(SCOPE>5) &&((POINTS AJUST.>73) XOR(TRANSACTIONS<=886))) &&(((MANAGER EXP.>0) NOR(MANAGER EXP.<=7)) XOR((TRANSACTIONS<=886) NOR(DURATION MONTHS<=36))))	62/62	4967.22580645	4344.61357266	15/15	14/15

Πίνακας Α.26: Λογικά Desharnais - Μέγεθος Δένδρου 5

A.6 Αποτελέσματα Κατηγορικών Εκτελέσεων – Αρχείο ISBSG

Execution	A/A	Tree	TRAINING SET			TESTING SET	
			Hit Ratio	Mean	Std	Hit Ratio	Success Ratio
Μέγεθος 2	1	((COMM='0') (RAD='0'))	374/374	6613.406417	11086.313146	92/93	85/92
Μέγεθος 2	2	((SYBASE='0') (DB2='0'))	374/374	6900.299465	13862.897223	92/93	87/92
Μέγεθος 2	3	((GOV='0') (INS='0'))	374/374	6983.425133	13806.514724	93/93	89/93
Μέγεθος 2	4	((RAD='0') (AEAU='0'))	374/374	6613.406417	11086.313146	92/93	85/92
Μέγεθος 2	5	((PADM='0') (MAN='0'))	374/374	6645.15508021	13412.8022825	93/93	86/93
Μέγεθος 2	6	((COMM='0') (PADM='0'))	374/374	6529.81818182	13217.5023760	93/93	87/93
Μέγεθος 3	1	((STDS='0') (IMS='0')) NAND((GOV='0') NOR(ORACLE='0'))	374/374	6613.406417	11086.313146	92/93	85/92
Μέγεθος 3	2	(IF ((SQLSRV='0') THEN ((ODT='0')) ELSE ((FPBS='0')))) IF ((RESL='1') THEN ((REGT='0')) ELSE ((VSAM='0'))))	374/374	6781.681818	13548.037597	93/93	86/93
Μέγεθος 3	4	((ACCESS='0') NOR(DB2='0')) NAND((ODT='0'))	374/374	6705.532085	12950.805370	93/93	85/93
Μέγεθος 3	5	((GOV='0') NOR(AEAU='0'))	374/374	6556.061497	13091.000419	93/93	85/93
Μέγεθος 3	6	((RAD='0') ((GOV='0') (STDS='0')))	374/374	6613.40641711	11086.3131464	93/93	86/93
Μέγεθος 3	7	((WATERF='0') (EMODEL='0')) ((DMODEL='0') (AEAU='0'))	374/374	6724.55347594	13001.3580186	93/93	86/93
Μέγεθος 3	8	((OOD='0') (EGW='0')) IF ((SQLSRV='0') THEN ((COMM='0')) ELSE ((EMODEL='0')))	374/374	6107.76470588	10560.6115575	93/93	81/93
Μέγεθος 3	9	((MAN='0') ((OOA='0') &&(OOD='0')))	374/374	6255.12538226	11119.3459041	93/93	83/93
Μέγεθος 3	10	((BAM='0') ((OOT='0') (BANK='0')))	374/374	6126.62079511	10984.3948741	93/93	84/93
Μέγεθος 3	11	((FPBS='0') ((ORACLE='0') (ODT='0')))	374/374	6978.48623853	14333.0498019	93/93	87/93
Μέγεθος 3	12	((OOA='0') ((PMODEL='0') XOR(PPL='JAVA')))	374/374	6164.29051988	10350.8863524	93/93	83/93

Μέγεθος 4	1	(((IMS='0')) NAND(PADM='0') NOR(WRT='1')) (((SYBASE='0') NOR(FPBS='0'))))	374/374	6613.406417	11086.313146	93/93	86/93
Μέγεθος 4	2	(((IMS='0')) NAND(PADM='0') NOR(WRT='1')) (((SYBASE='0') NOR(FPBS='0'))))	374/374	6726.98930481	13511.1842260	93/93	87/93
Μέγεθος 4	3	(((VSAM='0') NOR(AEAU='0')) NOR(EGW='0') (SYBASE='0')) NAND(IF (AEAU='0') THEN (VSAM='0') ELSE (JAD='0')) &&(STDS='0'))	374/374	6645.64705882	13436.3381975	93/93	83/93
Μέγεθος 4	4	(((VSAM='0') NAND(FPBS='0')) &&((IYEAR='2000') NOR(OOT='0')) NAND(((WRT='1') NAND(DB2='1'))))	374/374	6726.989304	13511.184226	93/93	87/93
Μέγεθος 4	5	((IF ((GOV='0') THEN ((DMODEL='0') ELSE ((OOA='0')) ((STDS='0') &&(PADM='0')) ((IF ((DMODEL='0') THEN (IYEAR='1998')) ELSE ((OOA='0')) &&((MAN='1') XOR(AEAU='0')))))	374/374	6613.40641711	11086.3131464	93/93	86/93
Μέγεθος 4	6	(((STDS='0')) XOR((IMS='0') XOR(OOT='0')) (((MAN='0') ((OOT='0') ((WRT='1')))))	374/374	7068.52139037	13827.9391961	93/93	90/93
Μέγεθος 4	7	(((INS='0') &&(OOD='0')) ((RAD='0') (STDS='0')) (((IYEAR='1998') (BAM='0')) XORIF ((STDS='0') THEN (EMODEL='0') ELSE ((ODT='0')))))	374/374	6664.90909091	13051.6028447	93/93	84/93
Μέγεθος 4	8	(((ADABAS='0') ((OOT='1') ((AEAU='0') (INS='0')) (((ORACLE='0') (ACCESS='0')) &&(TRST='0') (SYBASE='0'))))	374/374	6870.48663102	13589.4260638	93/93	86/93
Μέγεθος 4	9	(((JAD='0') (PADM='1') ((SQLSRV='0') (TBOX='0')) IF (IF (EGW='0') THEN (ADABAS='0') ELSE (EMODEL='0')) THEN (VSAM='0')) ELSE ((TRST='0'))))	374/374	6333.72994652	12447.4187073	93/93	83/93
Μέγεθος 4	10	(((O1DBS='0') (IMS='0') IF (BAM='0') THEN (O1DBS='0') ELSE (FPBS='0')) ((VSAM='0'))	374/374	6550.21122995	13096.8089276	93/93	85/93
Μέγεθος 5	1	((IF ((O1DBS='1') THEN (SQLSRV='0') ELSE (STDS='0')) NOR((RESL='1') &&(WRT='1')) NOR(((DB2='0') NOR(VSAM='0')))) NANDIF (IF ((EMODEL='0') XOR(MTEAM='0')) THEN ((PMODEL='0') NOR(WATERF='0')) ELSE ((SYBASE='0') XOR(BAM='0')) THEN (((DMODEL='0') (ORACLE='1') ((SYBASE='0') &&(SQLSRV='0')) ELSE (IF ((PADM='0') (STDS='0')) THEN ((TBOX='0')) ELSE ((WATERF='0') NAND(SYBASE='0'))))))))	374/374	7115.24642857	11964.2630377	93/93	87/93
Μέγεθος 5	2	(((IYEAR='2000') XOR(DMODEL='1') ((FPBS='0') NAND(MAN='0')) (((REGT='1') NOR(ODT='0')) (((IMS='1') &&(SQLSRV='0')) XORIF ((PMODEL='0') THEN (ODT='0') ELSE (WATERF='0')))))	374/374	6645.647058	13436.338197	93/93	83/93
Μέγεθος 5	3	(((RAD='0') &&(DB2='0')) &&((MAN='0') NAND(OOT='0')) NAND((((SQLSRV='0') NOR(INS='0')) &&((BANK='1') NAND(JAD='0')) &&(IF (BAM='0') THEN (O1DBS='1') ELSE (TBOX='0')) NANDIF ((TRST='0') THEN (AEAU='0') ELSE (BANK='0')))))	374/374	6943.21071429	12435.5425384	93/93	85/93
Μέγεθος 5	4	(((WRT='0') &&(EGW='0') ((MAN='1')) (((ADABAS='0') &&(INS='0')) &&(AEAU='0') &&(DMODEL='0')) ((IF (JAD='0') THEN (PADM='0') ELSE (VSAM='0')) &&(IYEAR='1999') &&(IMS='0')) NOR(((OOD='0') NAND(PADM='0')) XOR((COMM='0') XOR(OOD='0'))))	374/374	6404.933155	10873.613111	93/93	86/93
Μέγεθος 5	5	IF (((FPBS='0') XOR(GOV='0')) (((PMODEL='0') XOR(PADM='0')) &&((BANK='0') NOR(ODT='0')) THEN (((ORACLE='0') &&(ODT='0')) IF ((TRST='0') THEN (GOV='0') ELSE (PMODEL='0')) ((PADM='0'))	374/374	7009.19642857	14347.4933754	93/93	87/93

		NAND(ACCESS='0') NOR((COMM='0') NAND(RAD='0')))) ELSE (((((BANK='0') &&(BANK='0')))))					
Μέγεθος 5	6	((IF ((WRT='1') THEN ((WRT='1')) ELSE ((JAD='1')))) NORIF ((AEAU='0') THEN ((WRT='1')) ELSE ((IMS='1')))) NAND(((WATERF='0') XOR((TRST='0')))) (((REGT='0') XOR((INS='0') XOR(SQLSRV='0')))))	374/374	6338.87857143	13197.5207492	93/93	84/93
Μέγεθος 5	7	((INS='0') (((ODT='0') NAND(EGW='0')) NORIF ((IMS='0') THEN ((DMODEL='0')) ELSE ((BANK='0')))) NOR(((SYBASE='0') ((MTEAM='0')) &&((DMODEL='1') NOR(ORACLE='0')))))	374/374	7069.54285714	14072.2640508	93/93	87/93
Μέγεθος 5	8	(((((OOT='1')) IF ((SYBASE='0') THEN ((INS='0')) ELSE ((MAN='0')))) NAND((((FPBS='0') NAND(MTEAM='1')) XORIF ((WRT='1') THEN ((OOA='0')) ELSE ((VSAM='0')))))	374/374	6724	14026.001522	93/93	86/93

Παράρτημα Β

Παρουσίαση Γραφικών Παραστάσεων Καλύτερων Εκτελέσεων

B.1 Εισαγωγή

B.2 Γραφικές Παραστάσεις Καλύτερων Εκτελέσεων

B.1 Εισαγωγή

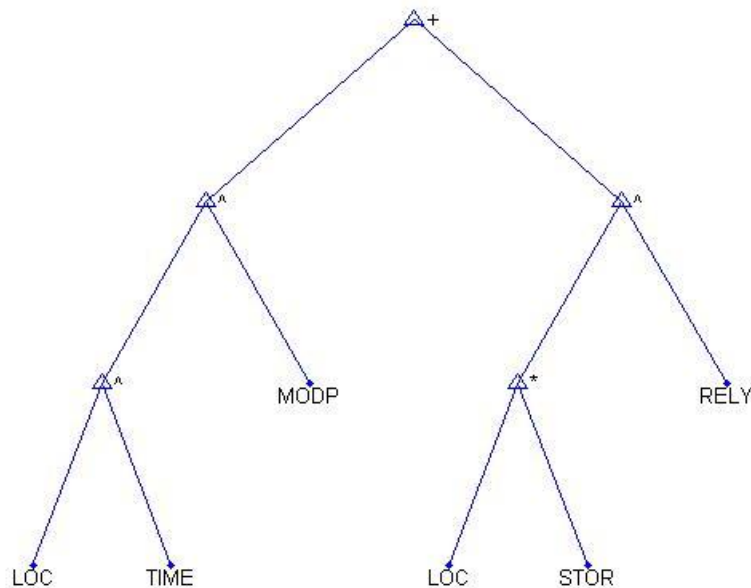
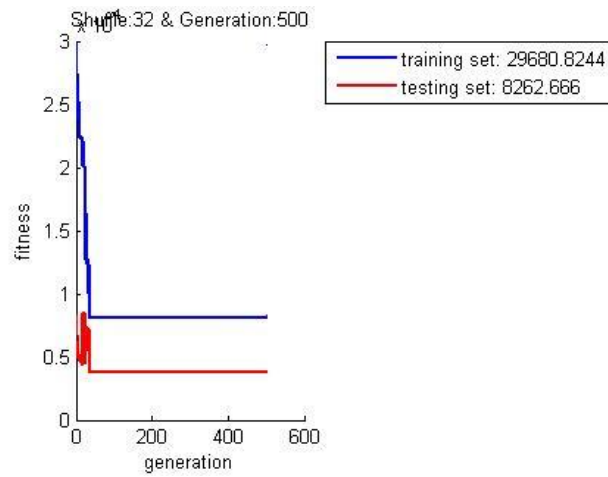
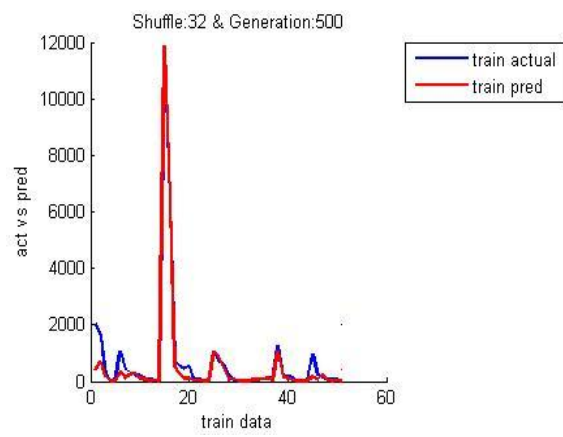
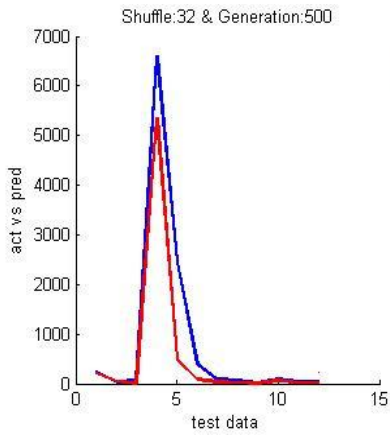
Σε αυτό το κομμάτι θα παραθέσουμε όλες τις γραφικές παραστάσεις για κάθε ένα από τα καλύτερα αποτελέσματα που έχουμε επιλέξει. Θα δοθούν οι γραφικές παραστάσεις του δένδρου της παραγόμενης εξίσωσης, η γραφική παράσταση του fitness και τέλος οι δύο γραφικές παραστάσεις του πραγματικού - υπολογίσιμου τόσο για τα δεδομένα ελέγχου όσο και για τα δεδομένα εκμάθησης. Αυτές οι γραφικές παραστάσεις ισχύουν στις αριθμητικές εκτελέσεις ενώ για τις λογικές και κατηγορικές εκτελέσεις θα έχουμε μόνο τη γραφική παράσταση του δένδρου του καλύτερου ατόμου. Όλες οι γραφικές παραστάσεις έχουν δημιουργηθεί από το προτεινόμενο σύστημα των γραφικών παραστάσεων όπως έχει περιγραφεί στην παράγραφο 4.3.5.

B.2 Γραφικές Παραστάσεις Καλύτερων Εκτελέσεων

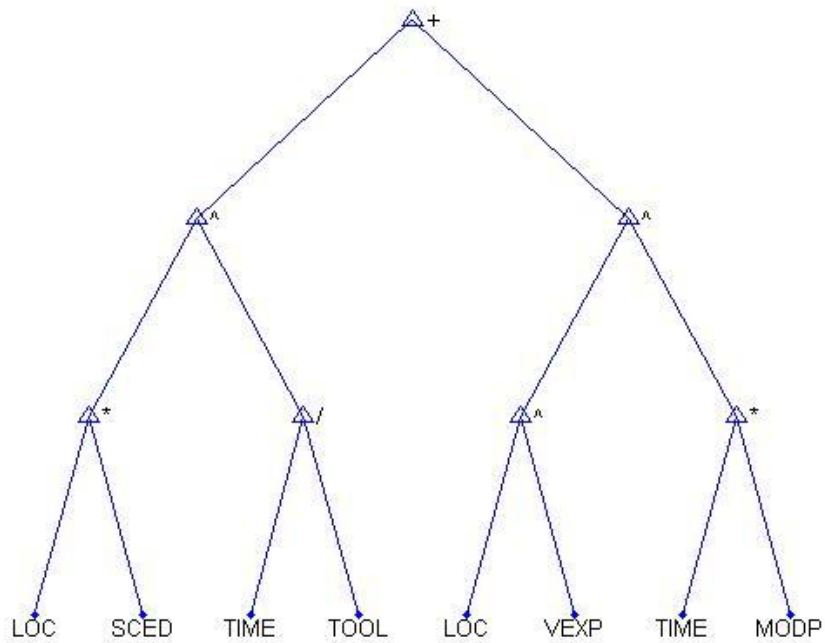
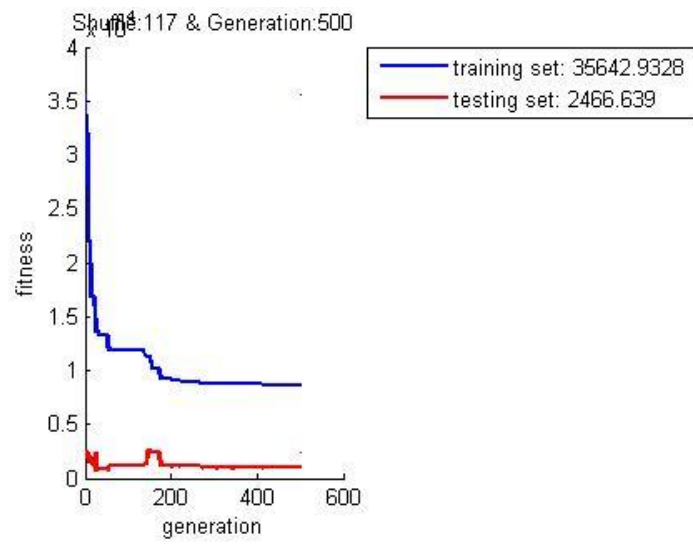
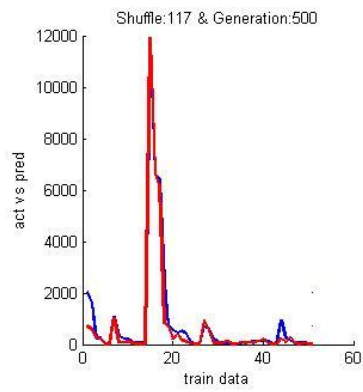
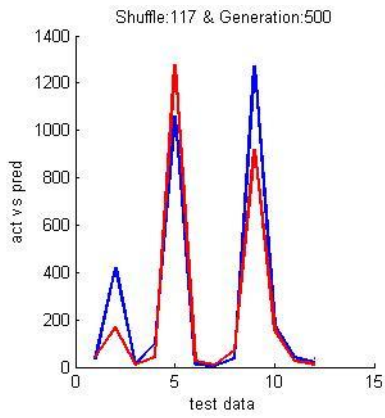
Θα παραθέσουμε τις παραστάσεις με σειρά σύμφωνα με τις εκτελέσεις που έχουν γίνει. Πρώτα θα δοθούν οι παραστάσεις για τις αριθμητικές εκτελέσεις με πρώτο το αρχείο δεδομένων Cocomo και δεύτερο με το αρχείο δεδομένων Desharnais. Στην συνέχεια με την ίδια σειρά όσο αφορά τα αρχεία θα παρατεθούν οι παραστάσεις για τις λογικές εκτελέσεις και τέλος για τις κατηγορικές. Πιο κάτω παρουσιάζονται κάποιες ενδεικτικές γραφικές παραστάσεις για κάποιες επιλεγμένες εκτελέσεις.

Β.2.1 Γραφικές Παραστάσεις Αριθμητικών Εκτελέσεων - Cosomo

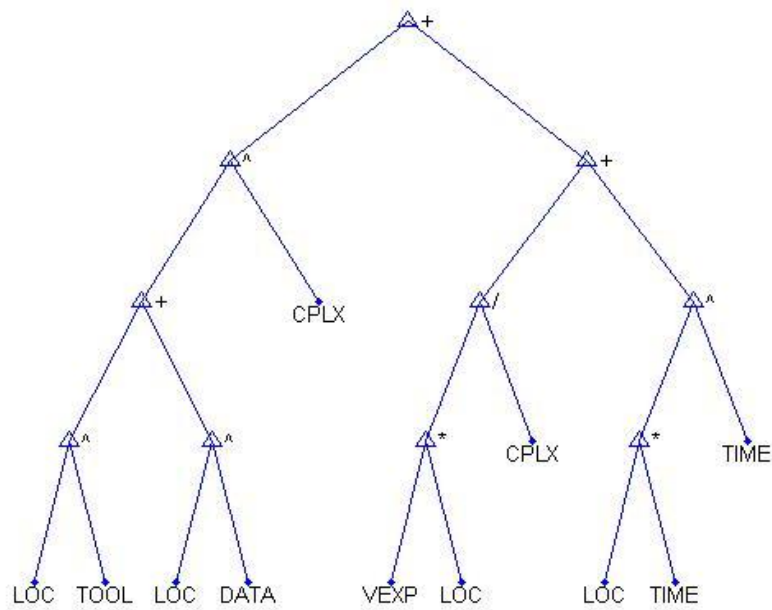
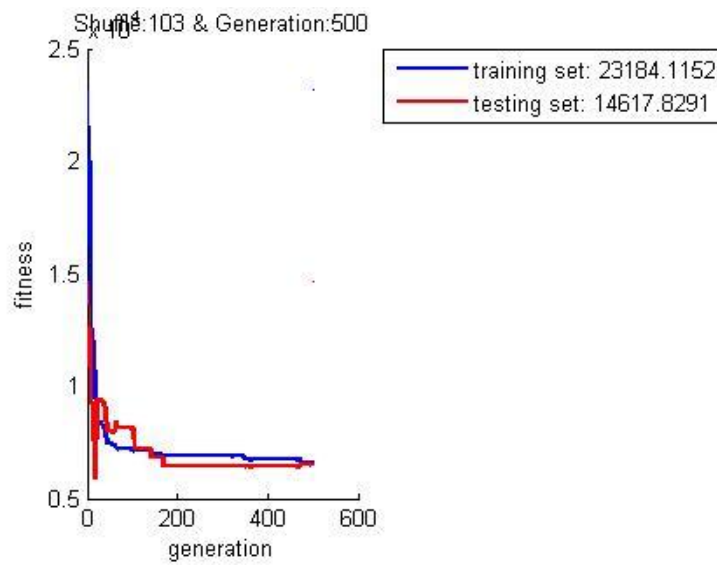
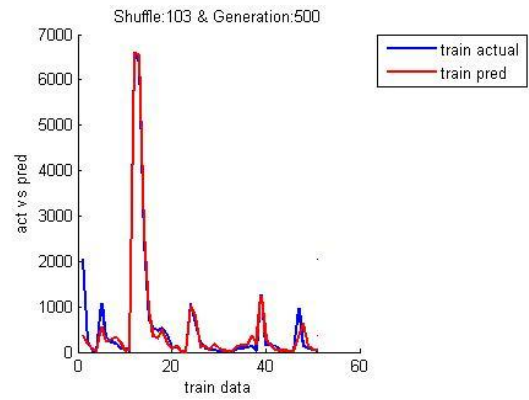
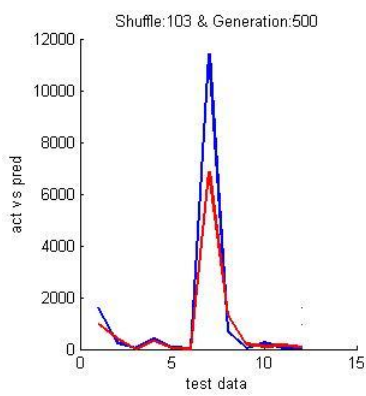
Εκτέλεση Α: Βάθος 4 - 32:



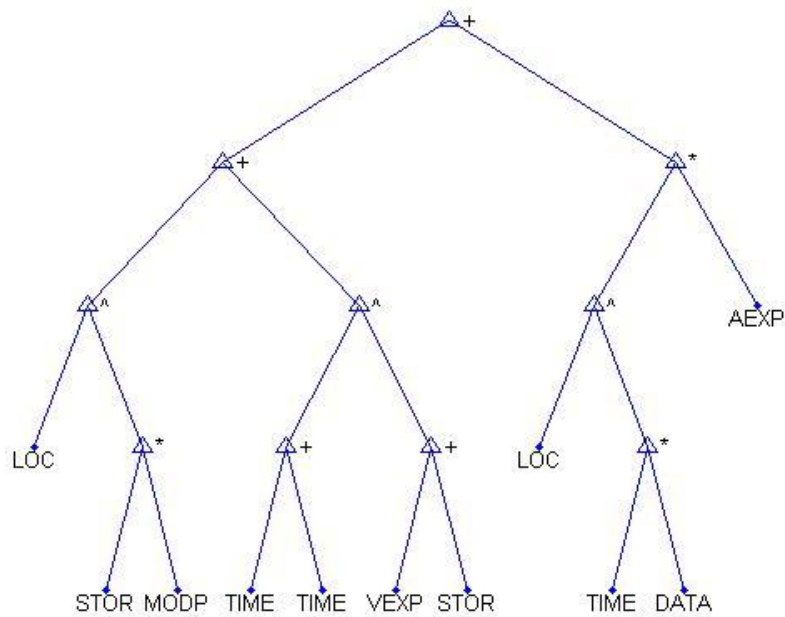
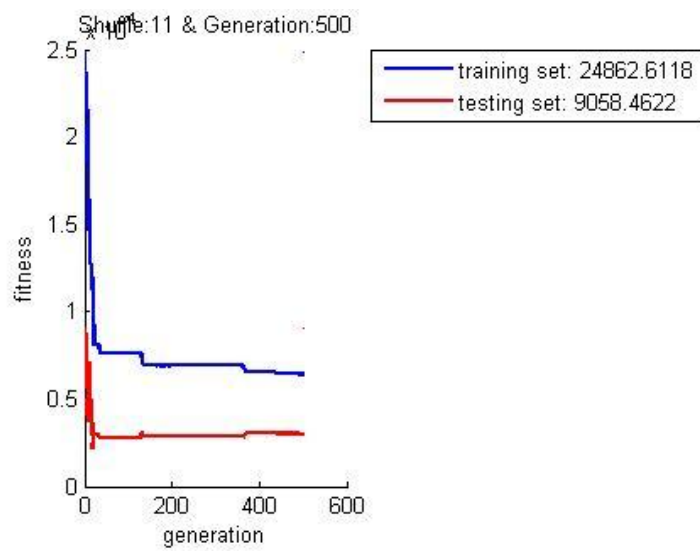
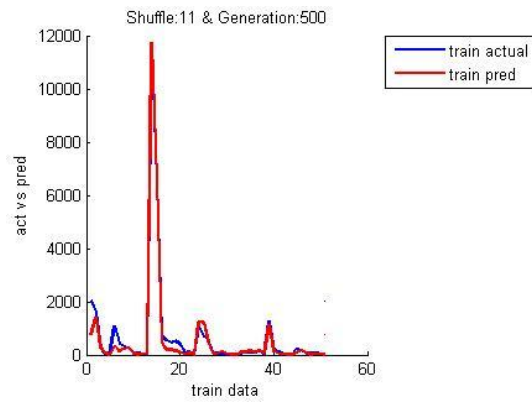
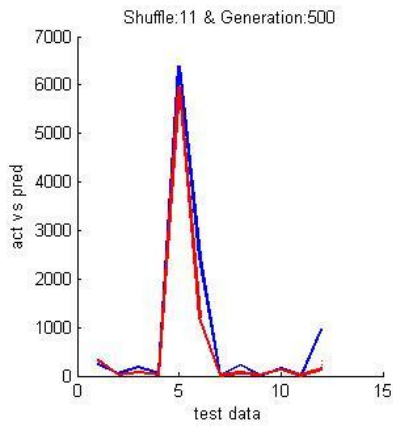
Εκτέλεση A: Βάθος 4 - 117:



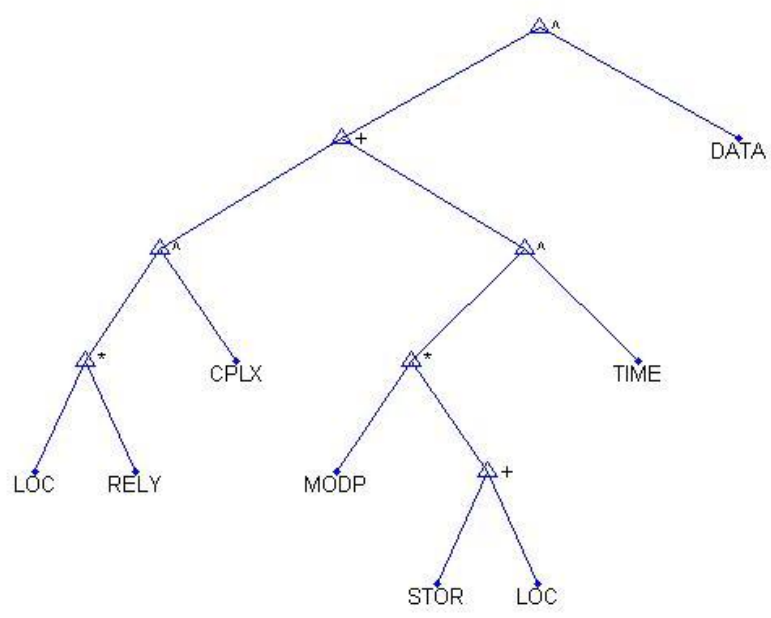
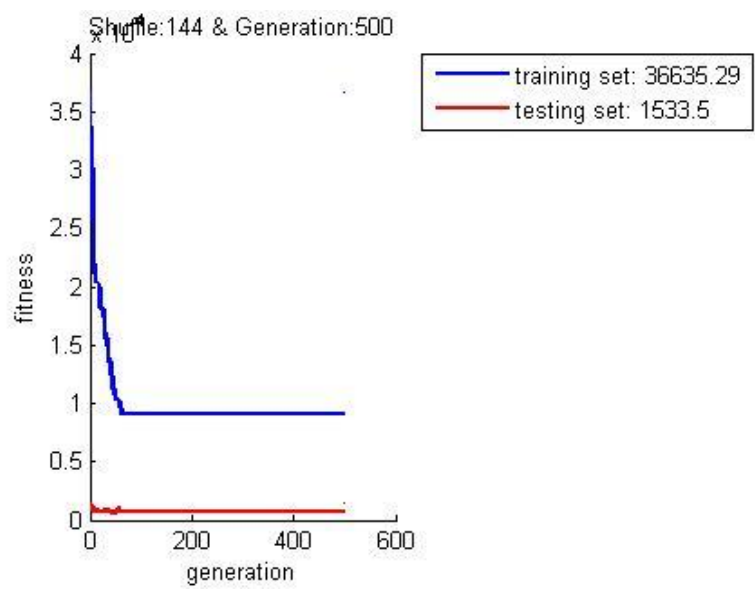
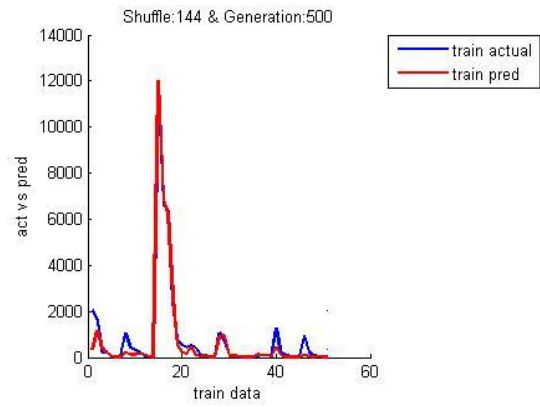
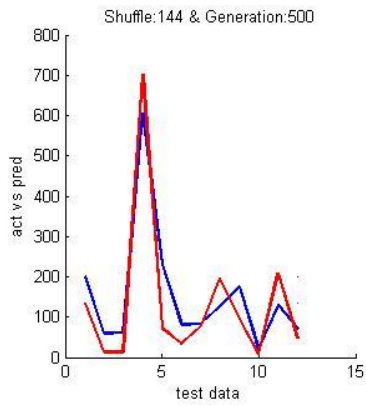
Εκτέλεση A: Βάθος 5 - 103:



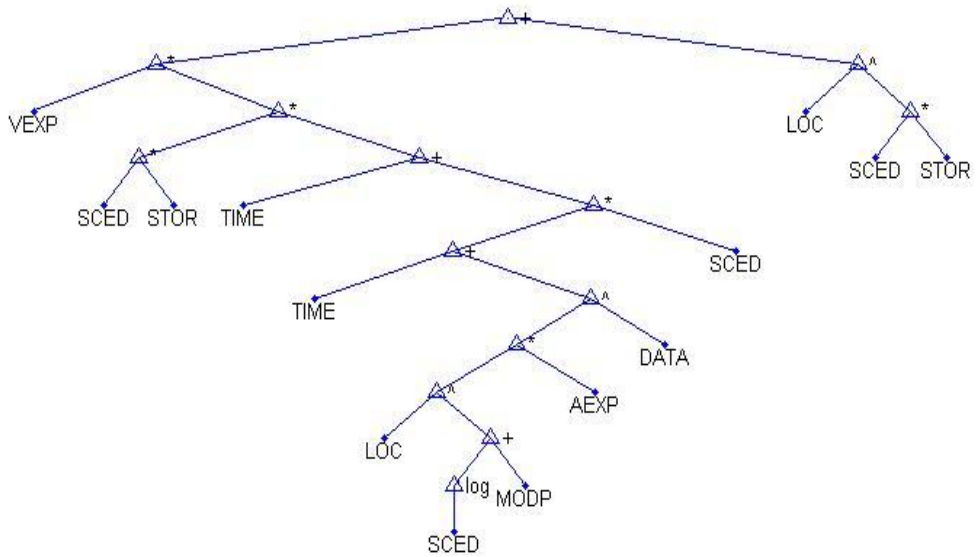
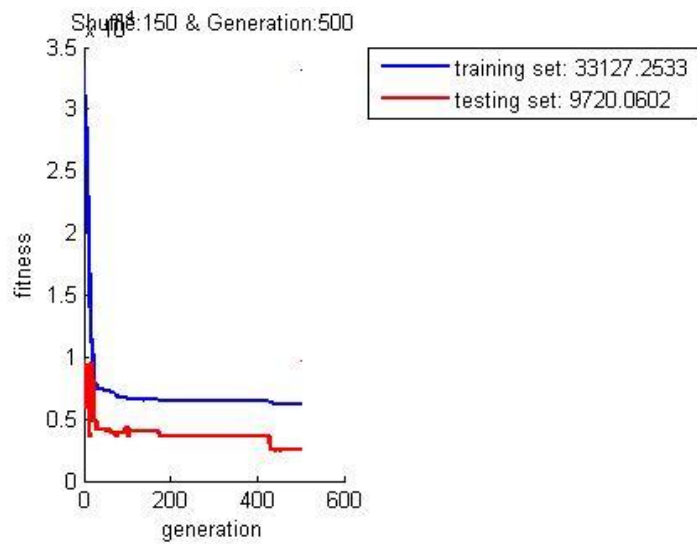
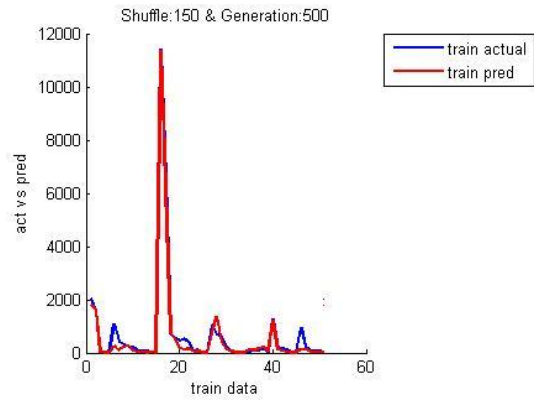
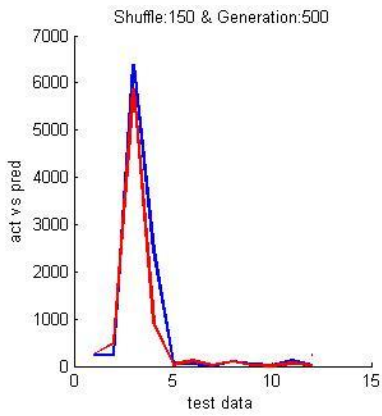
Εκτέλεση A: Βάθος 5 - 11:



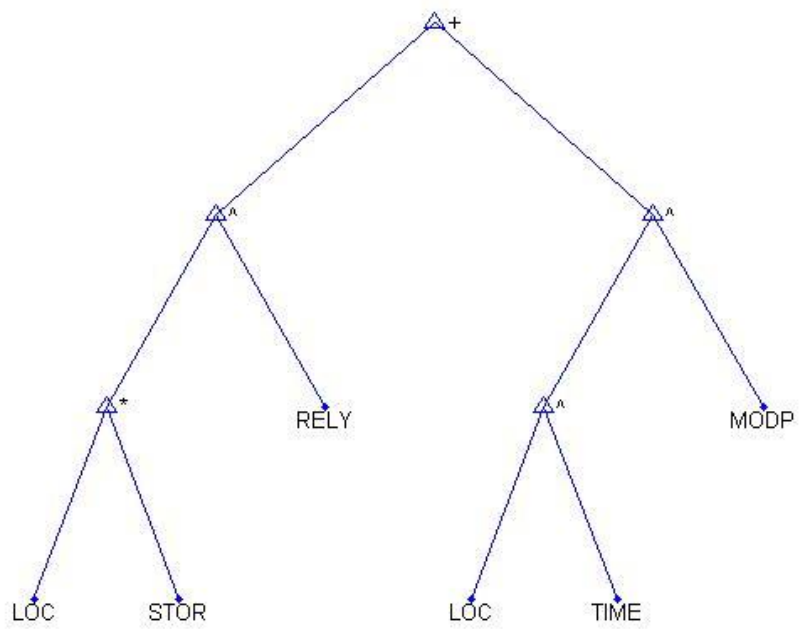
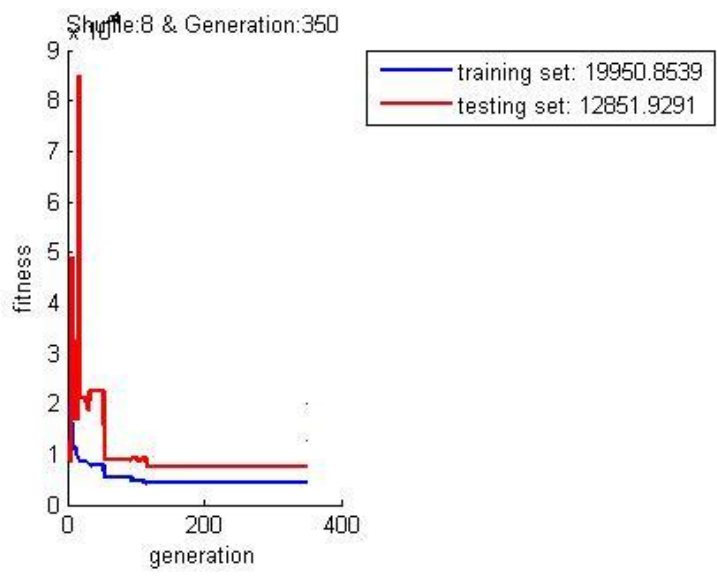
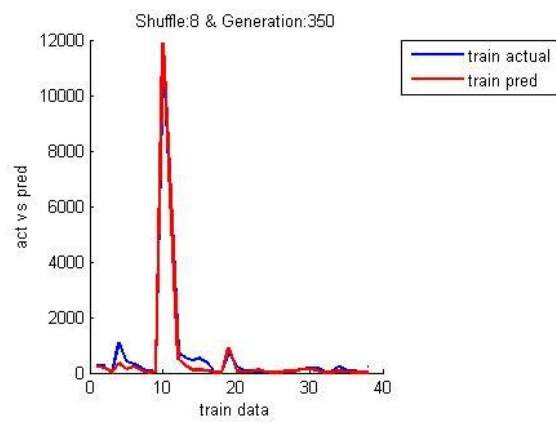
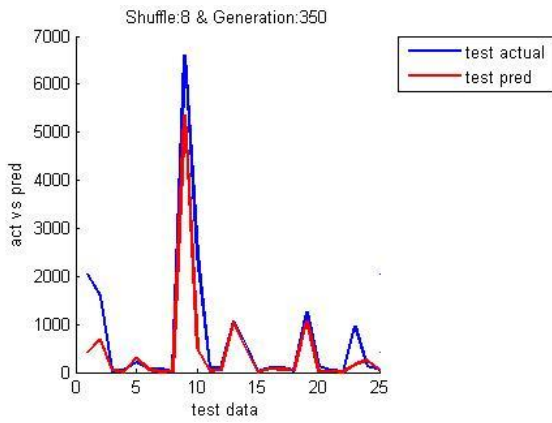
Εκτέλεση A: Κόμβοι 16 - 144:



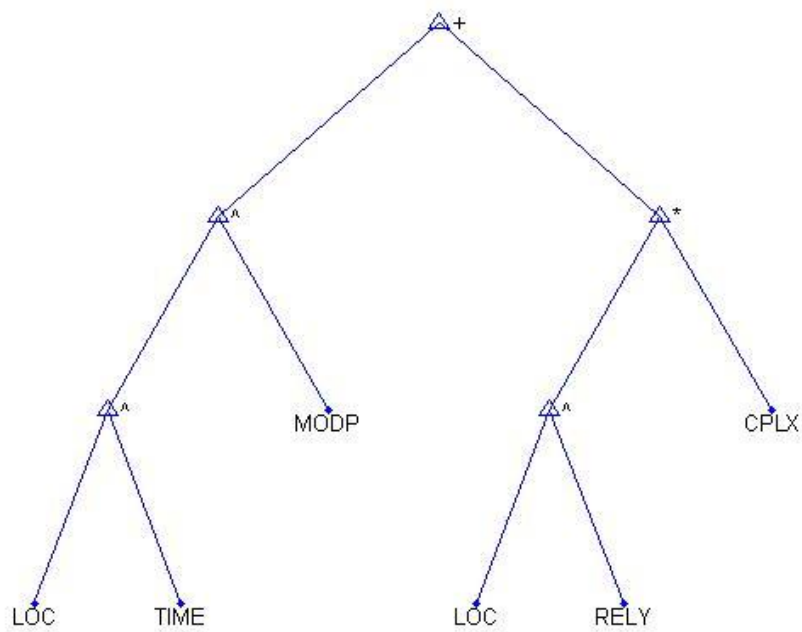
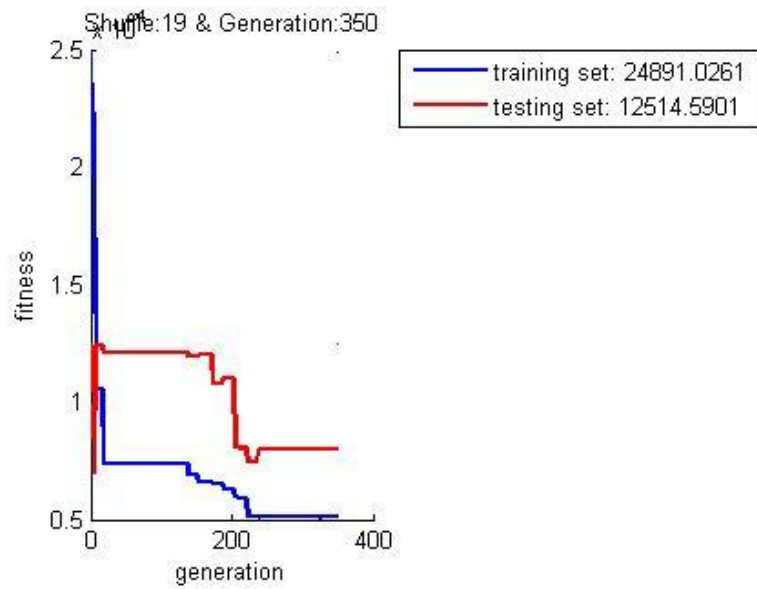
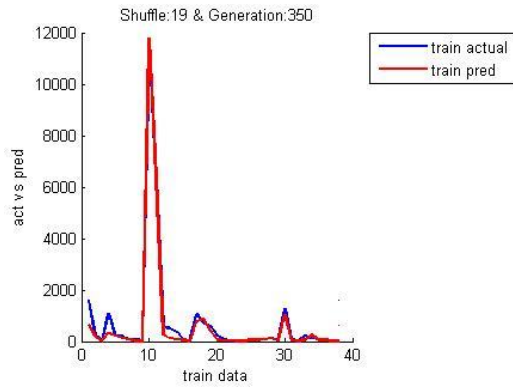
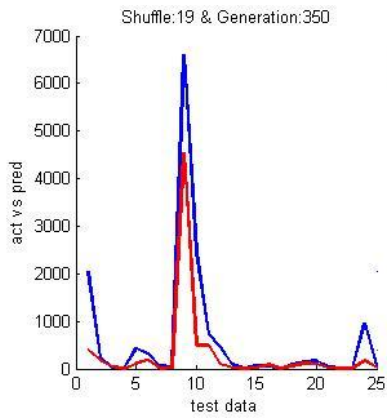
Εκτέλεση A: Κόμβοι 28 - 150:



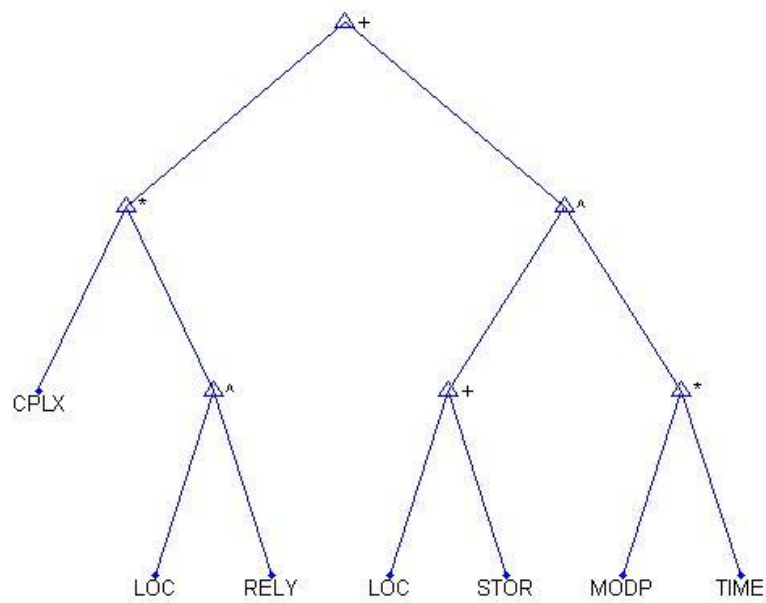
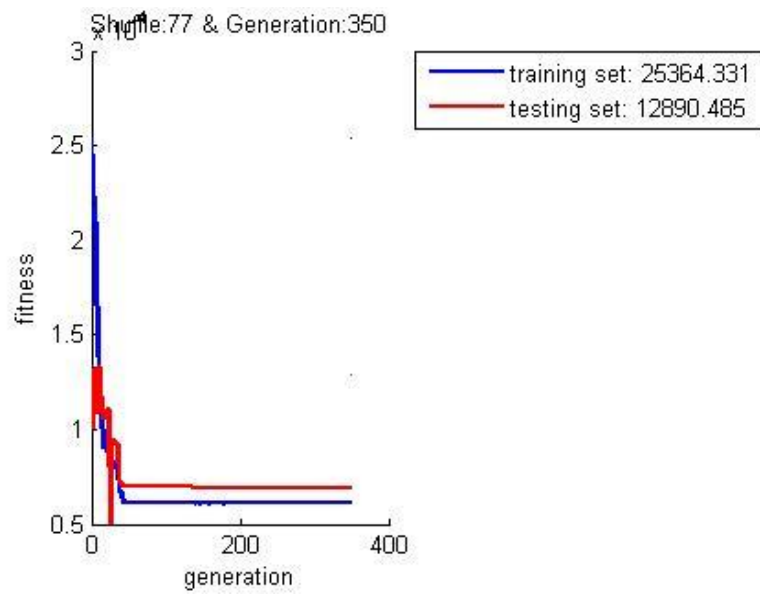
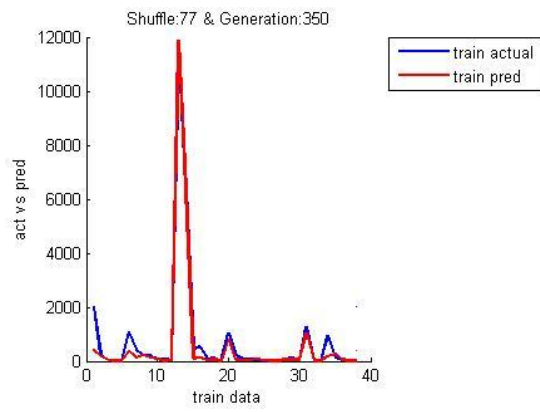
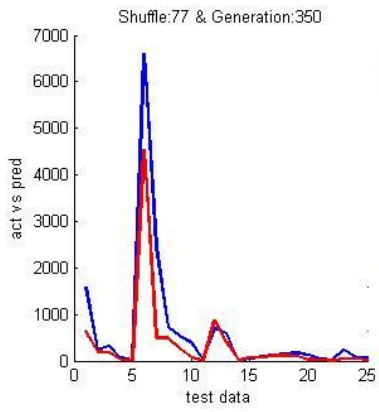
Εκτέλεση Β: Βάθος 4 - 8:



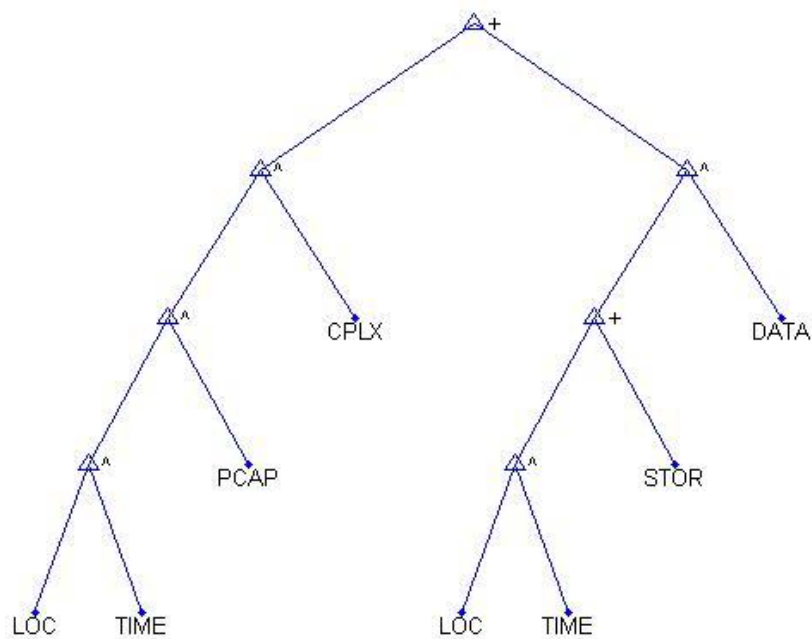
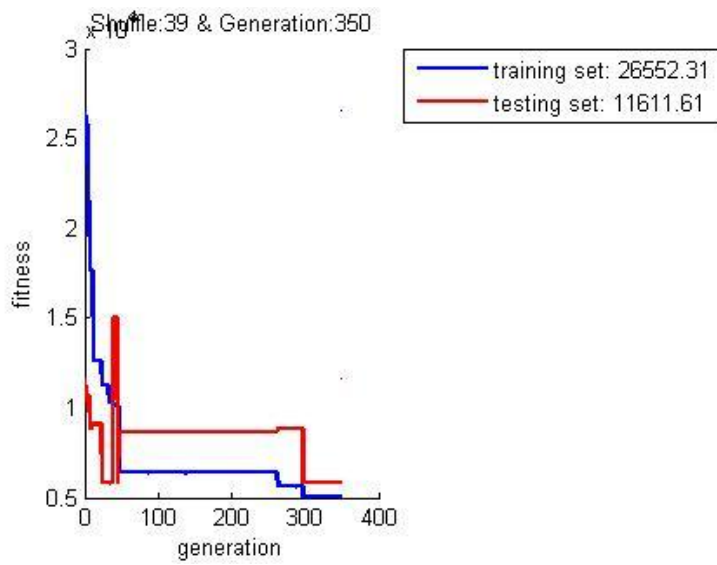
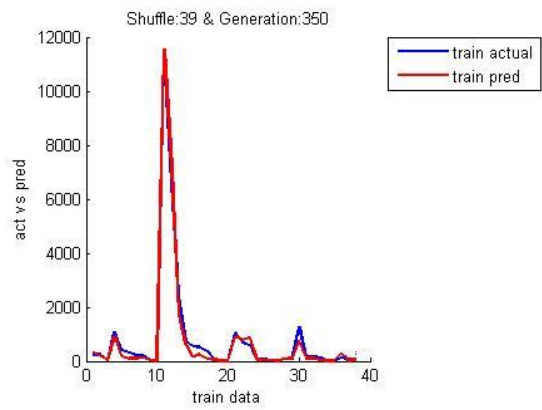
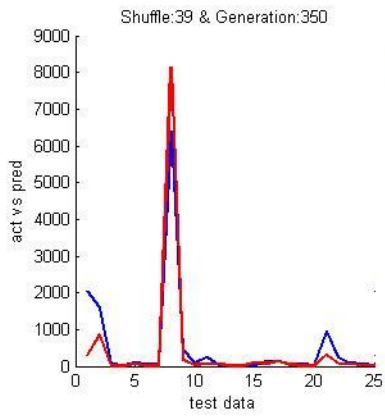
Εκτέλεση Β: Βάθος 4 - 19:



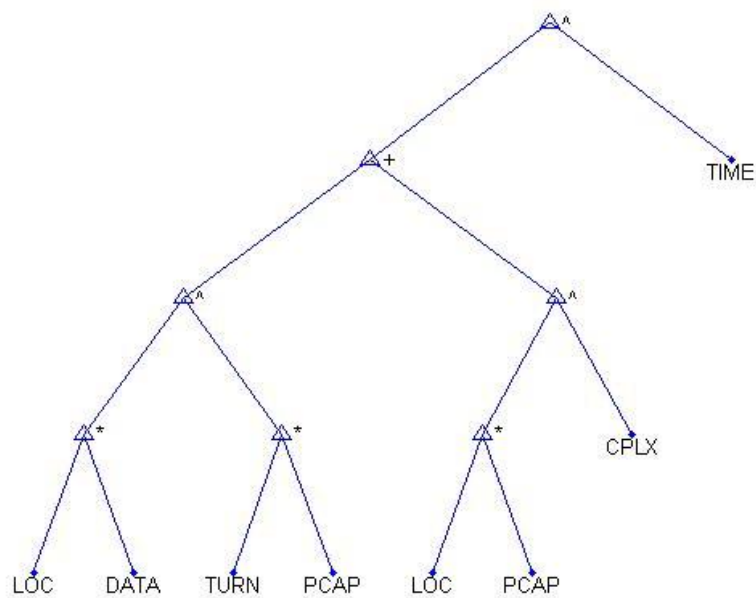
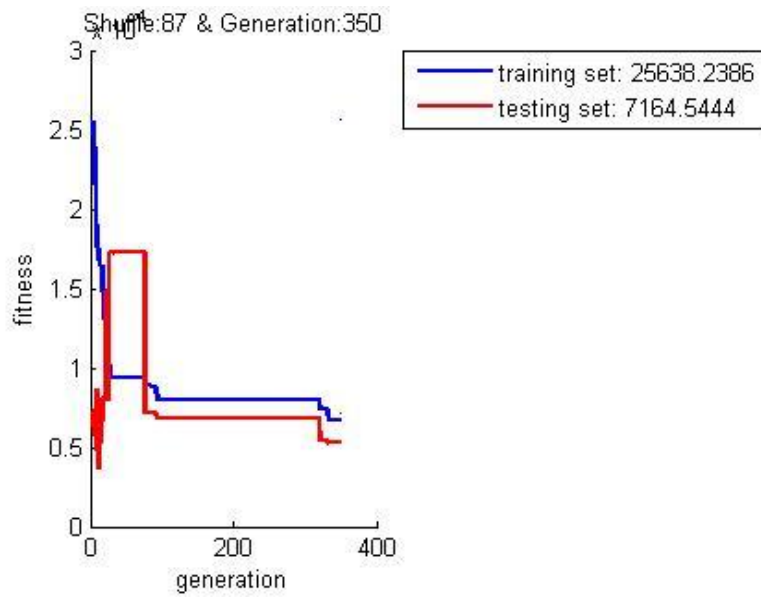
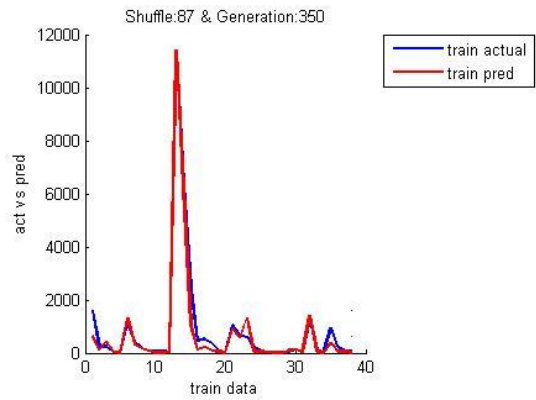
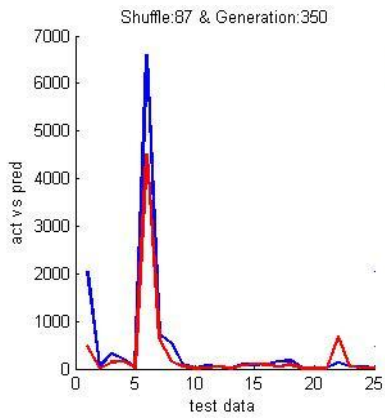
Εκτέλεση Β: Βάθος 4 - 77:



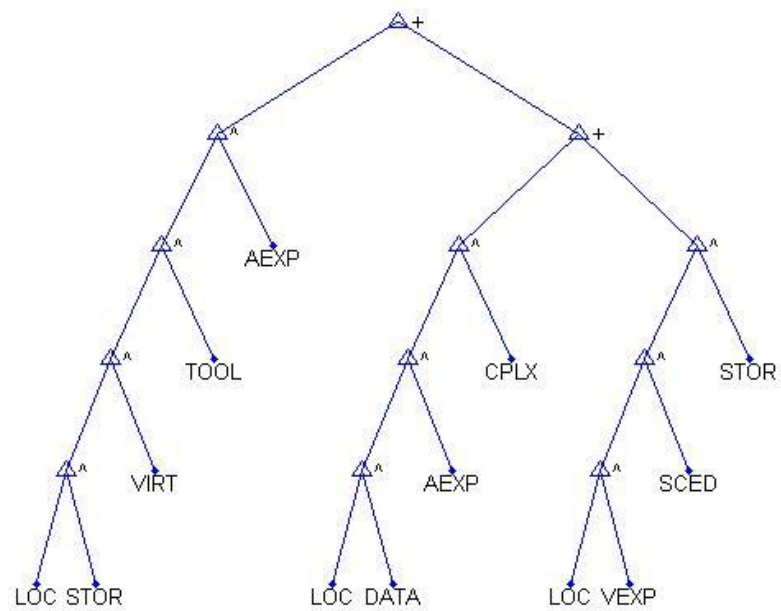
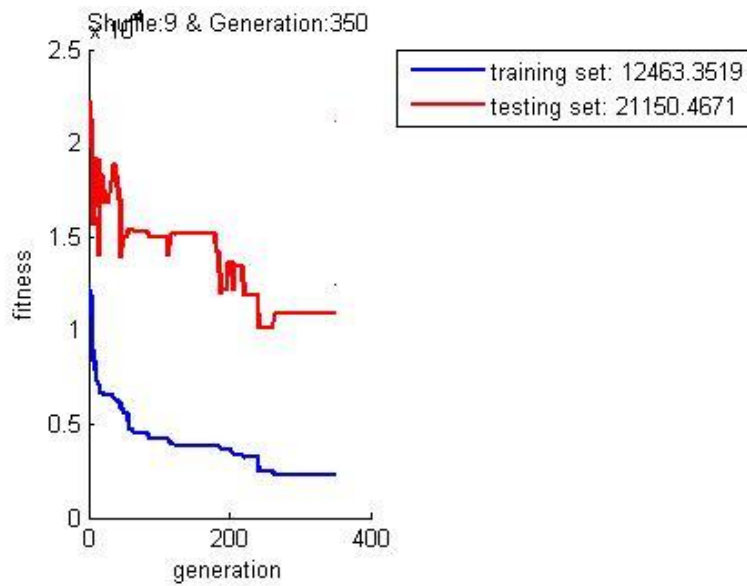
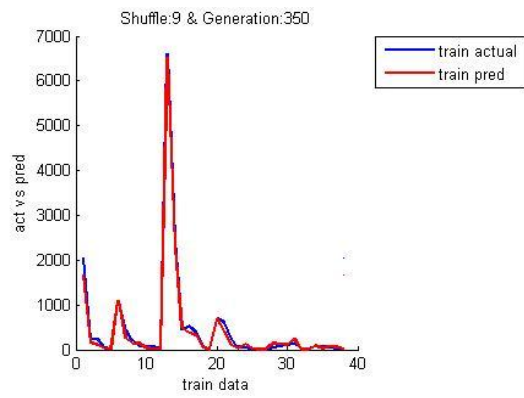
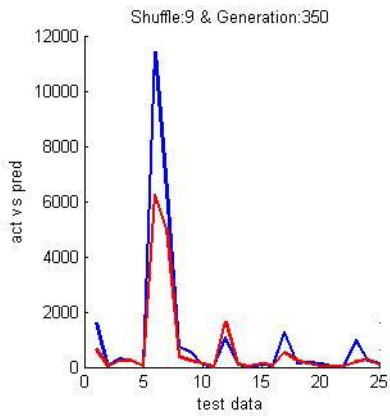
Εκτέλεση Β: Βάθος 5 - 39:



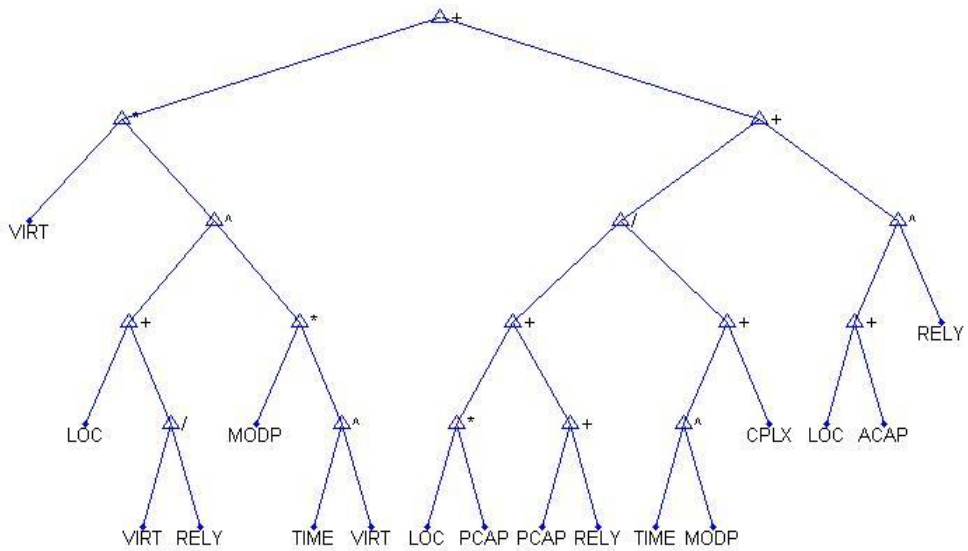
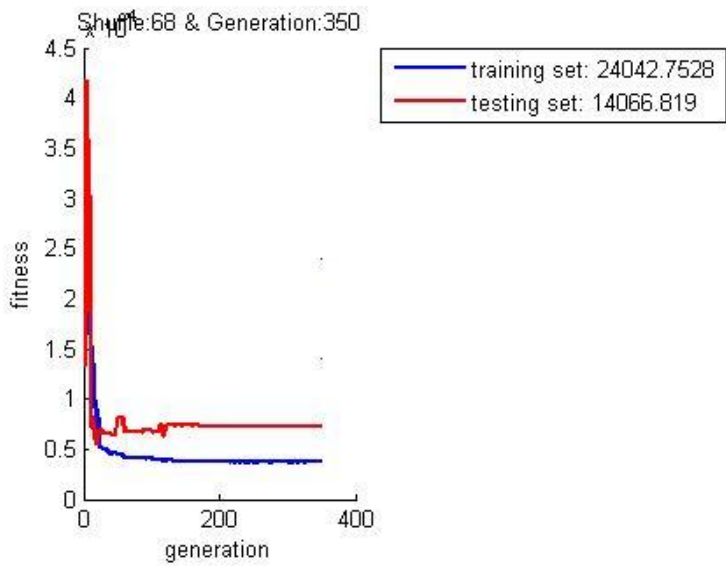
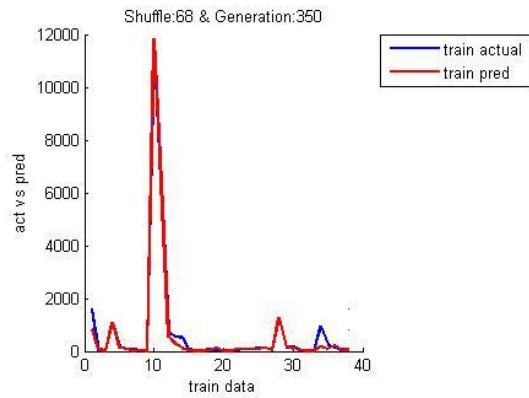
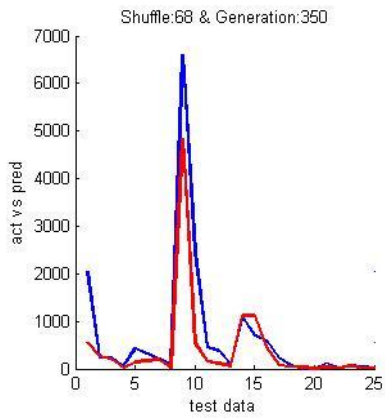
Εκτέλεση Β: Βάθος 5 - 87:



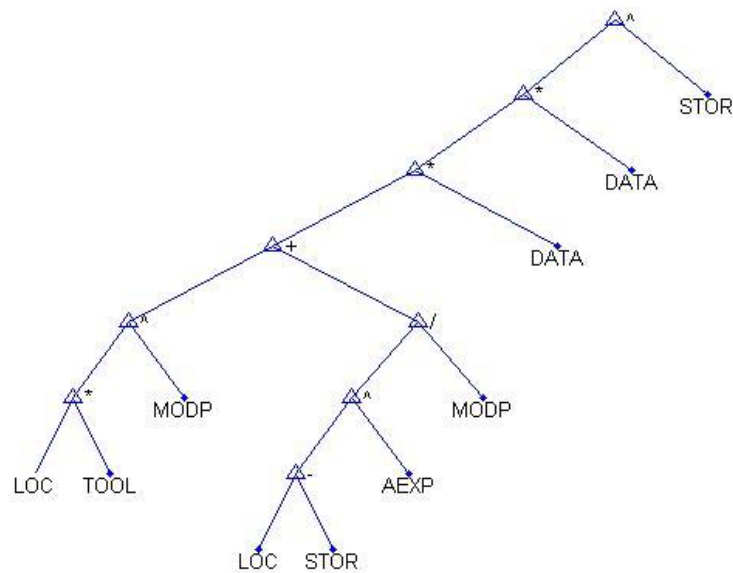
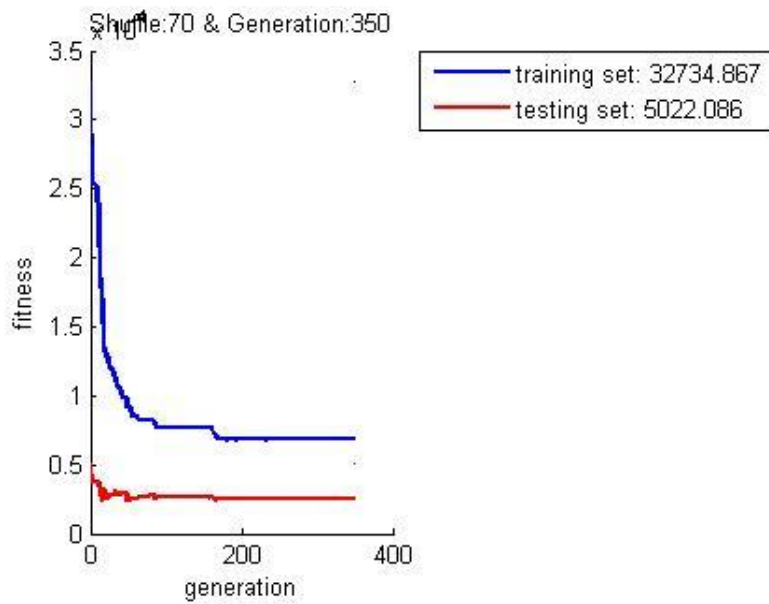
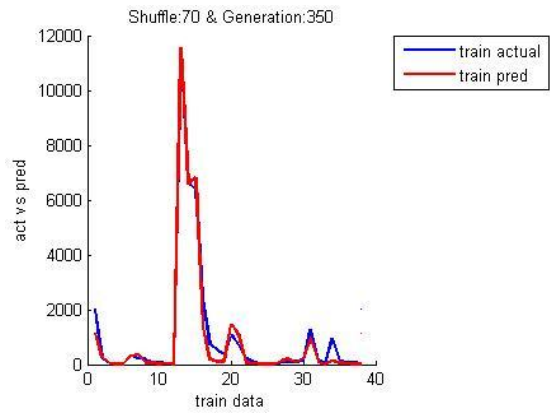
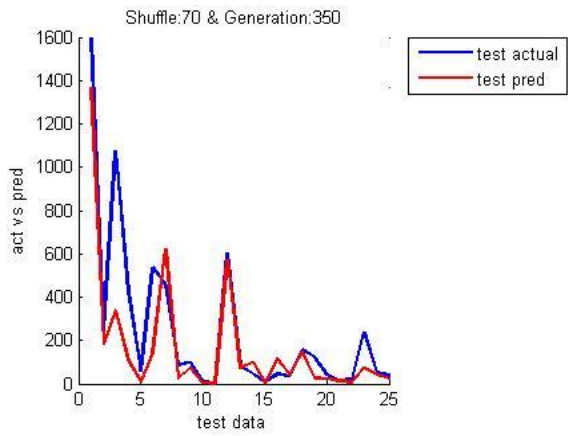
Εκτέλεση Β: Βάθος 6 - 9:



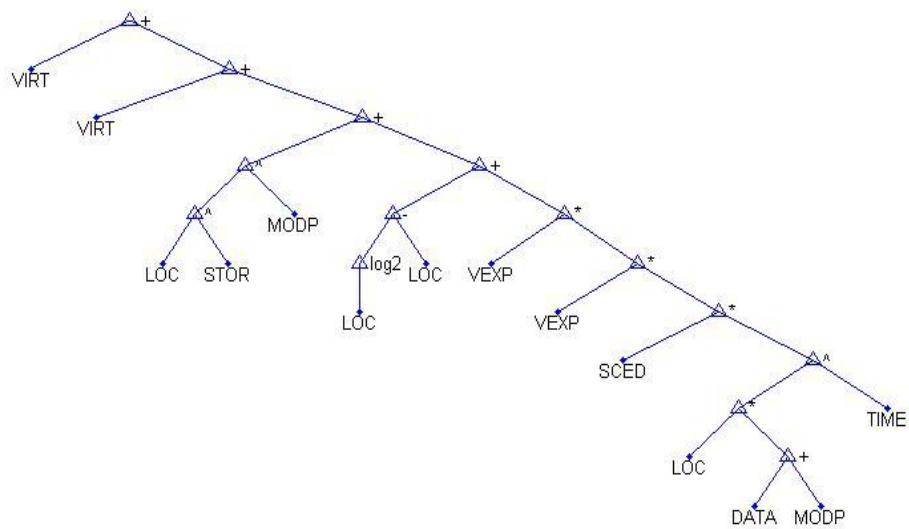
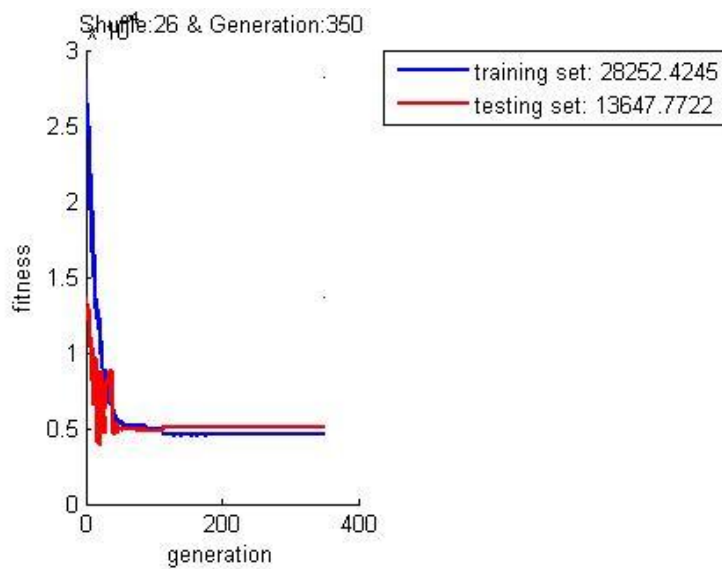
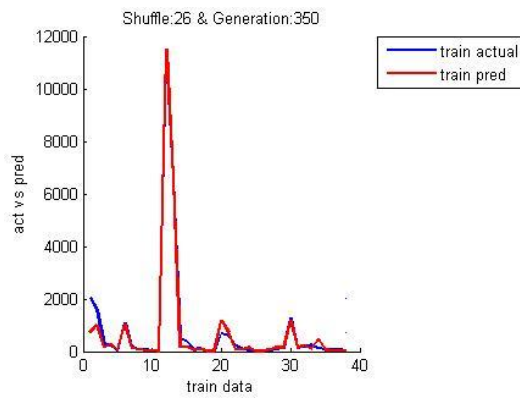
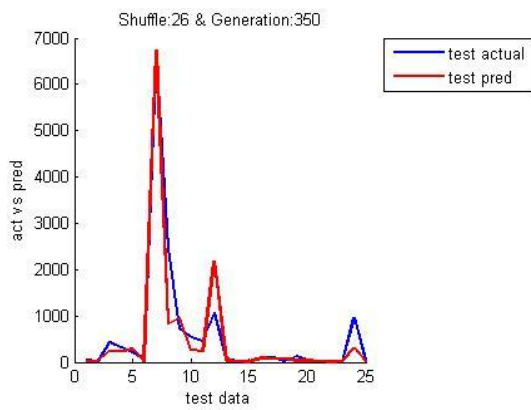
Εκτέλεση Β: Βάθος 6 - 68:



Εκτέλεση Β: Κόμβοι 20 - 70:

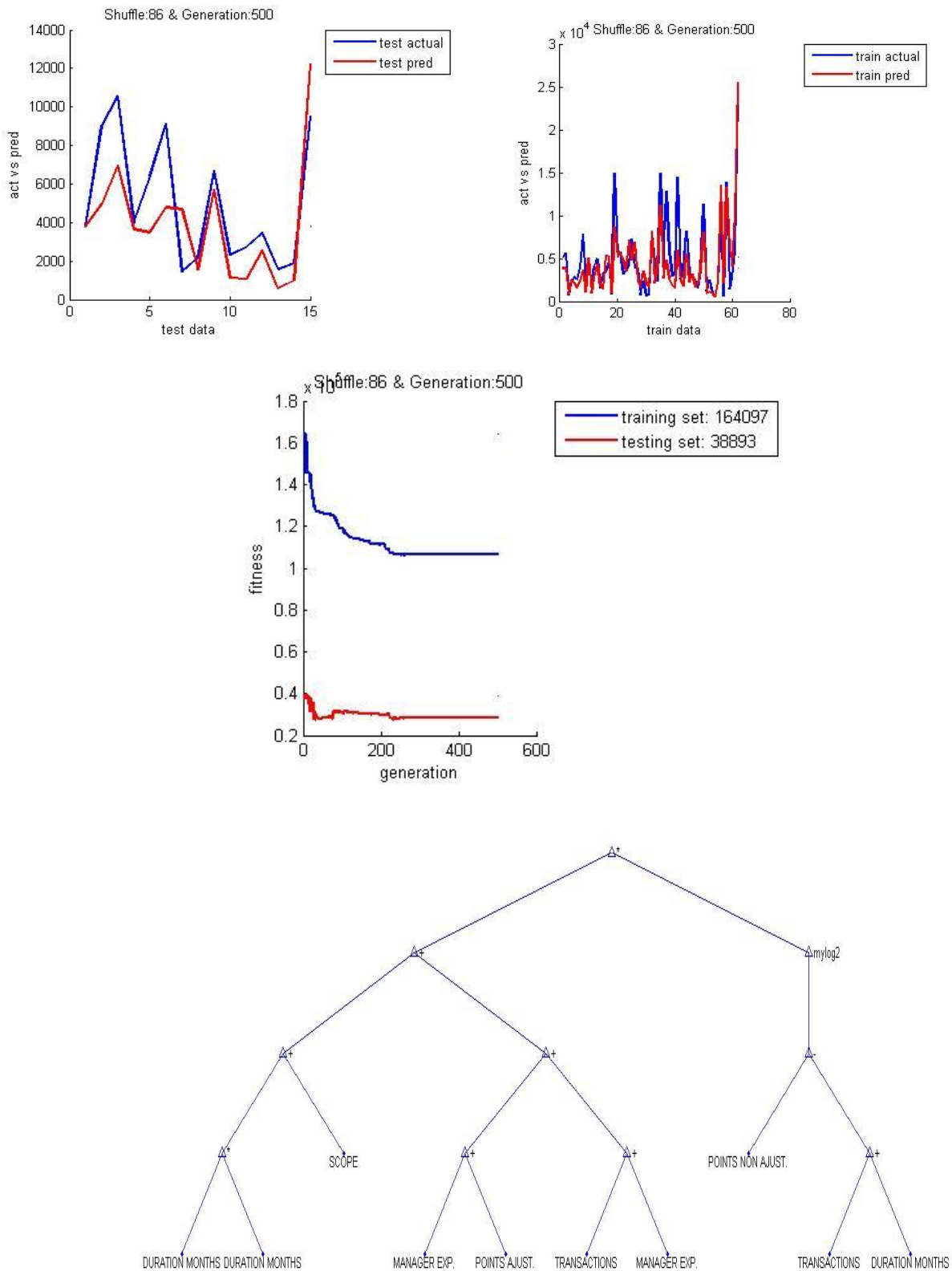


Εκτέλεση Β: Κόμβοι 28 - 26:

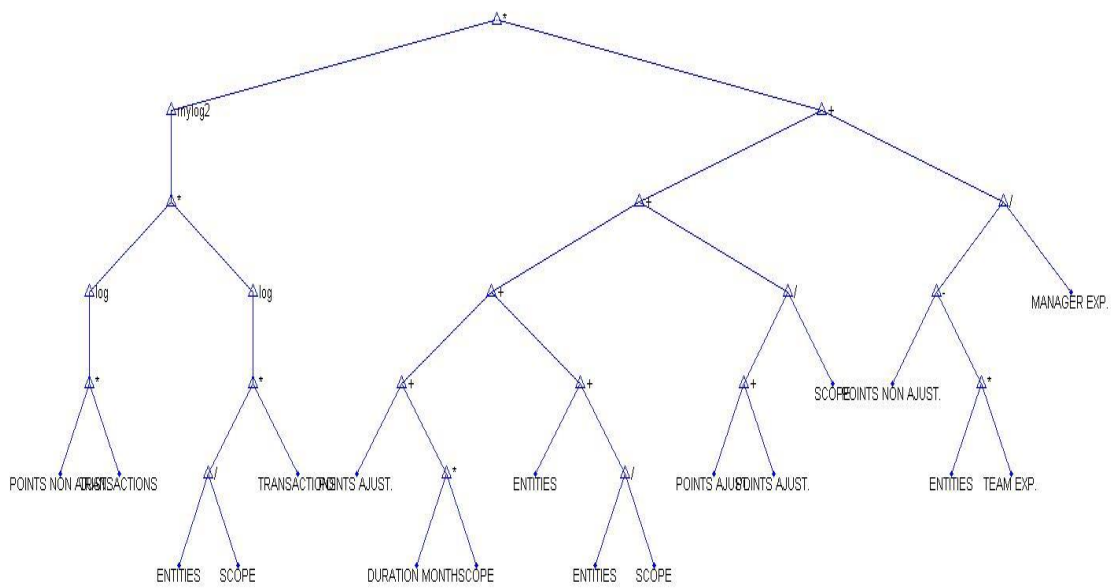
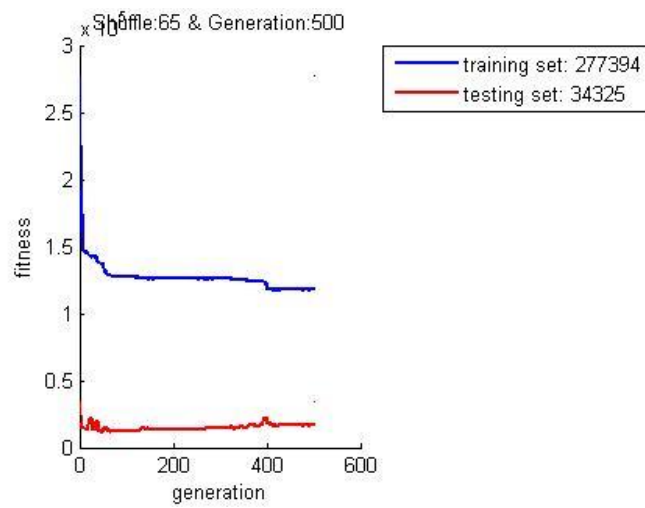
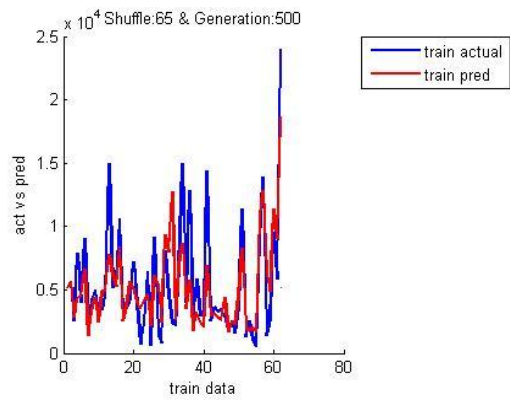
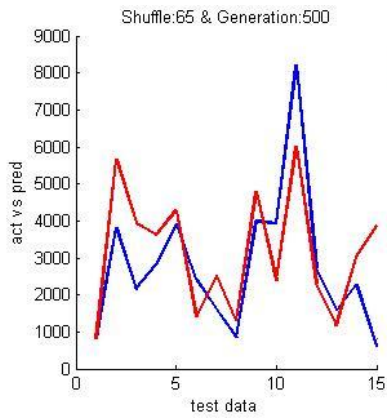


B.2.2 Γραφικές Παραστάσεις Αριθμητικών Εκτελέσεων - Desharnais

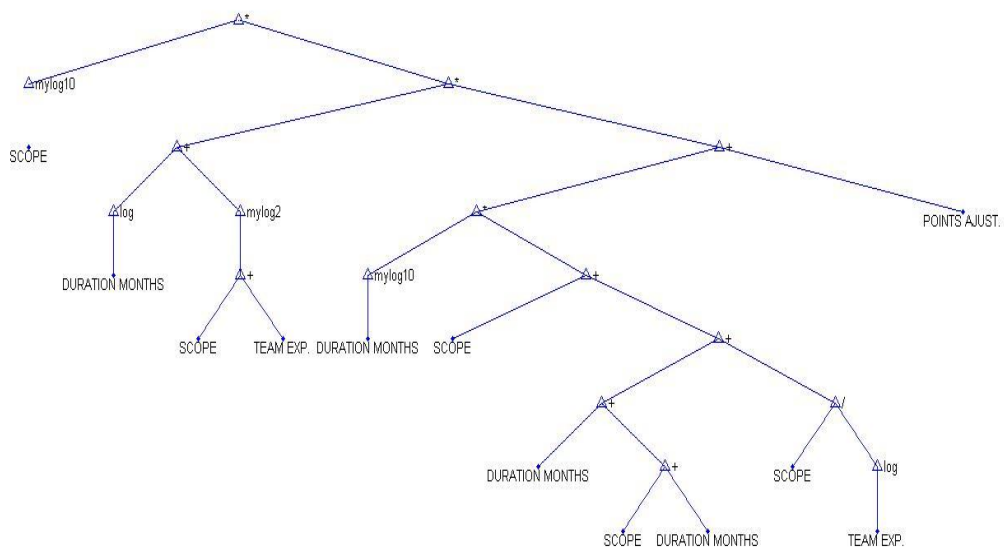
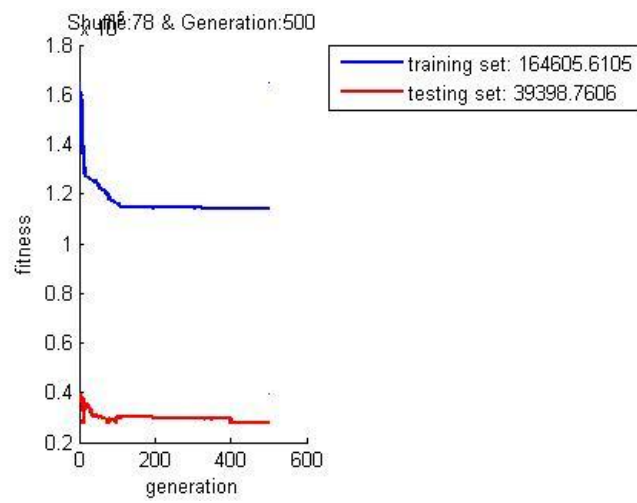
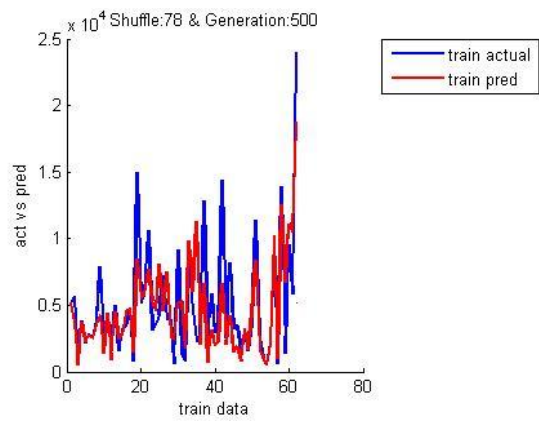
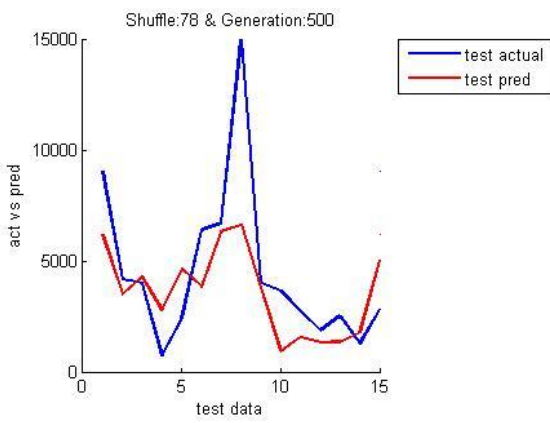
Εκτέλεση A: Βάθος 5 - 86:



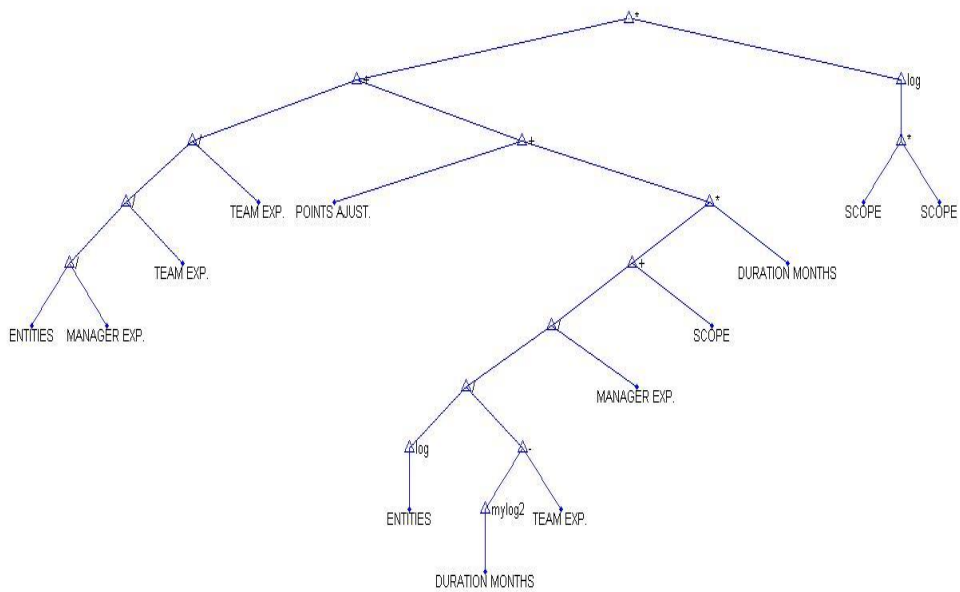
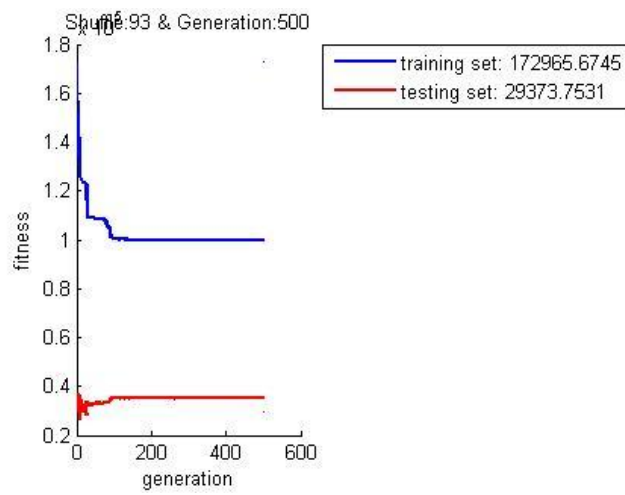
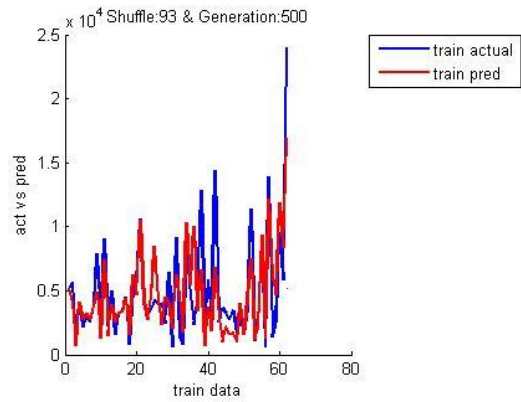
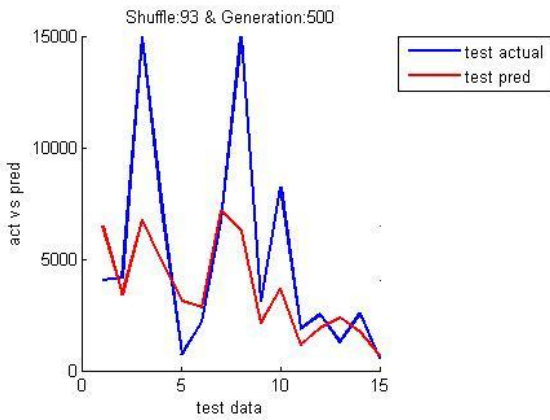
Εκτέλεση A: Βάθος 7 - 65:



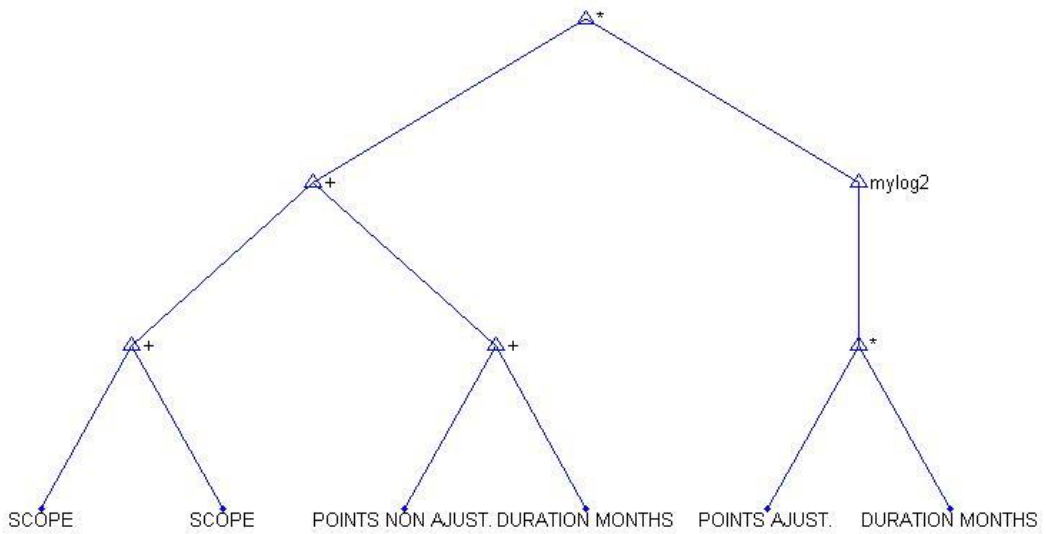
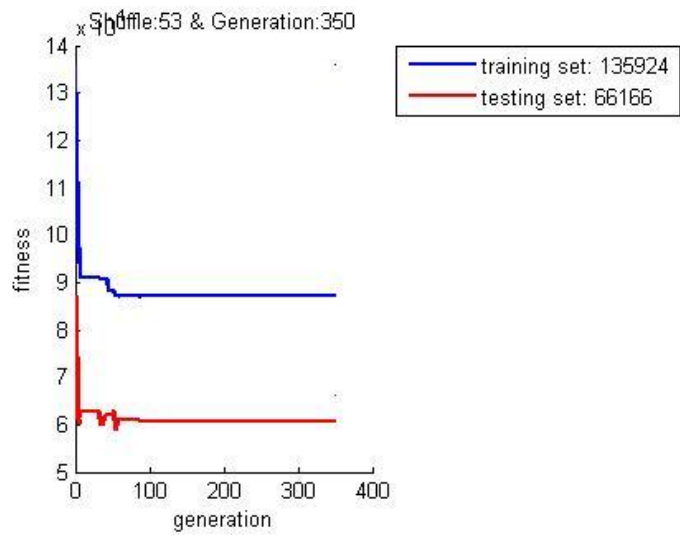
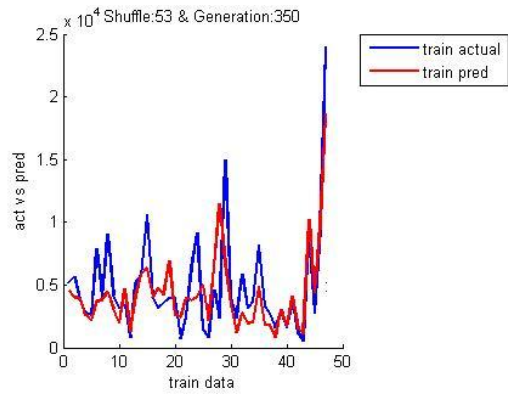
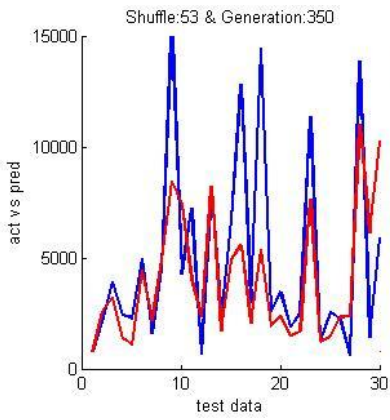
Εκτέλεση A: Κόμβοι 28 - 78:



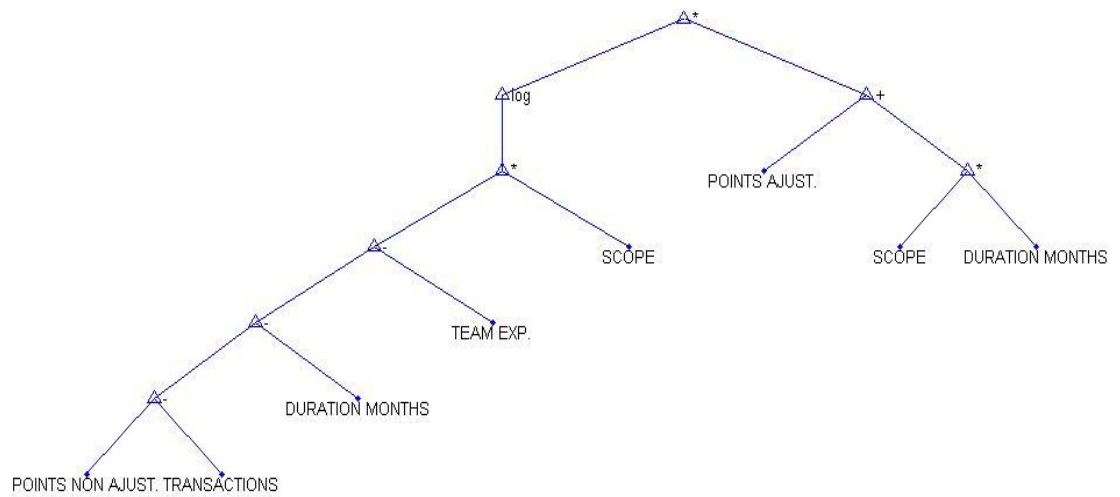
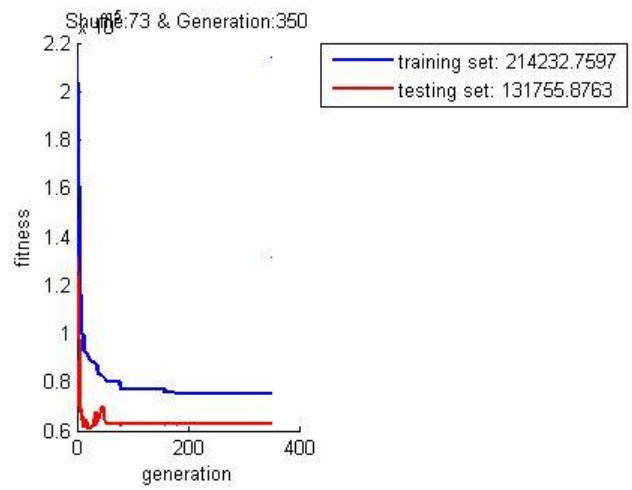
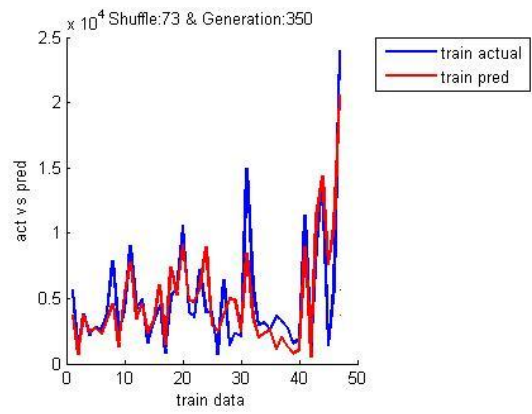
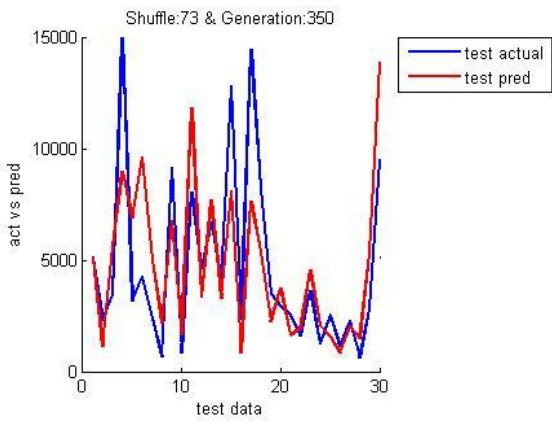
Εκτέλεση A: Κόμβοι 28 - 93:



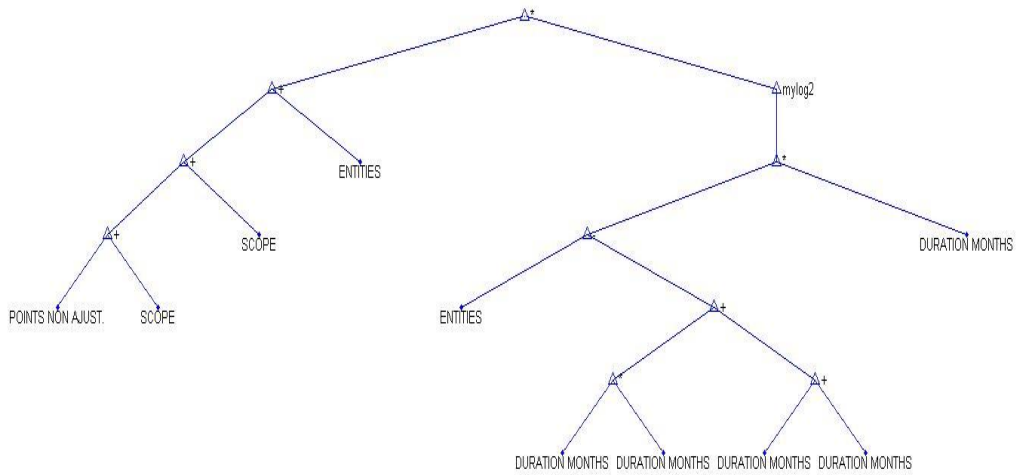
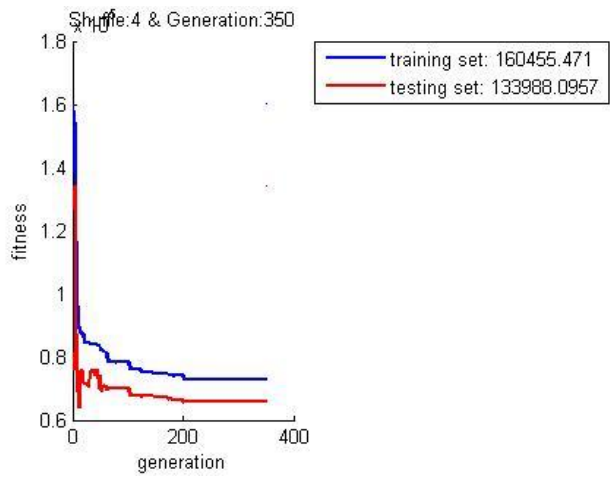
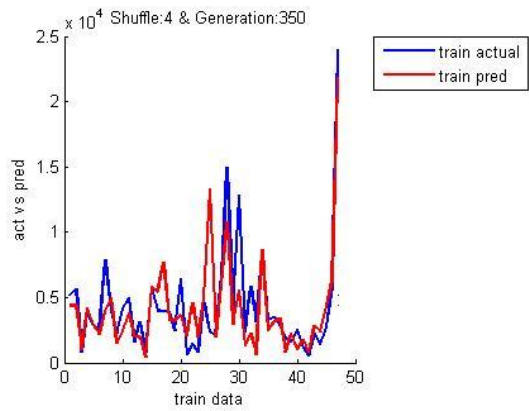
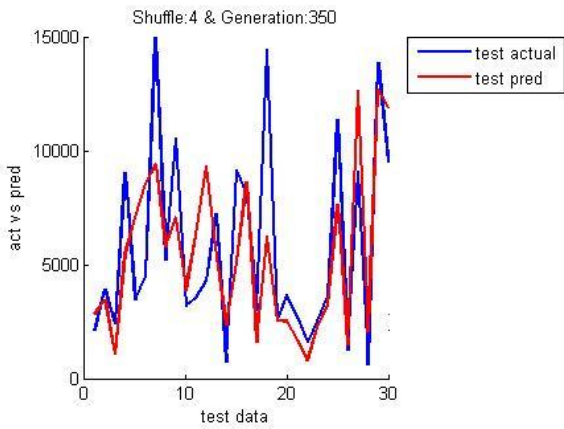
Εκτέλεση Β: Βάθος 4 - 53:



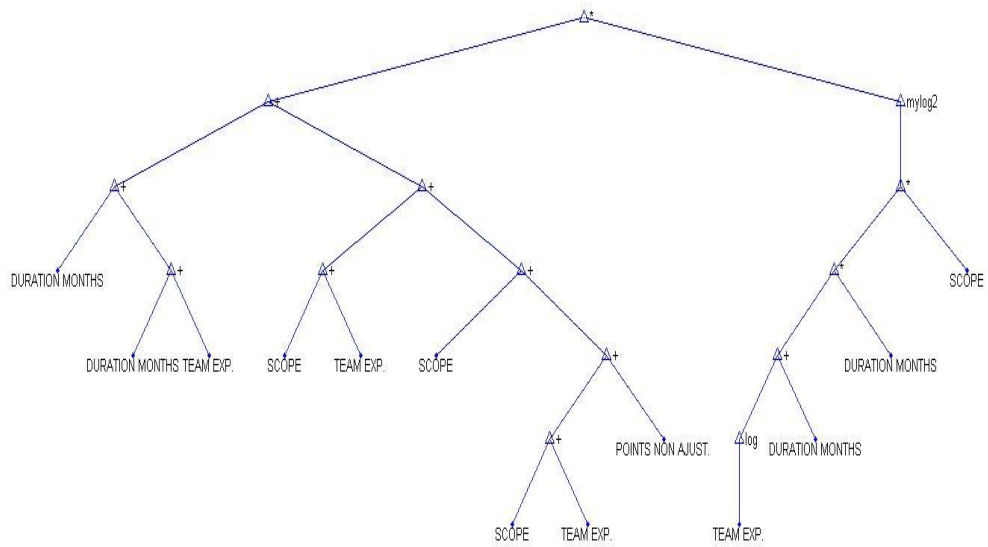
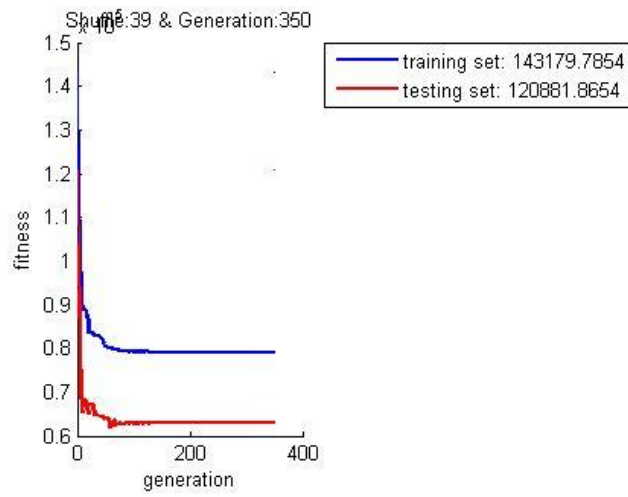
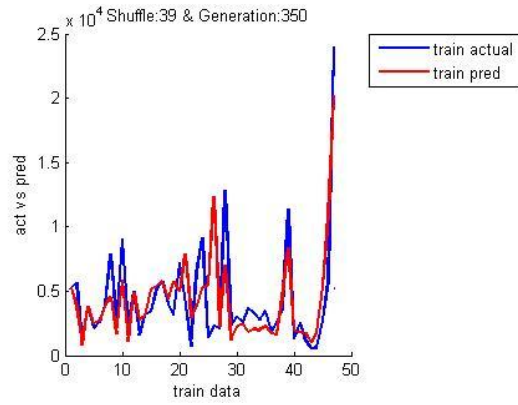
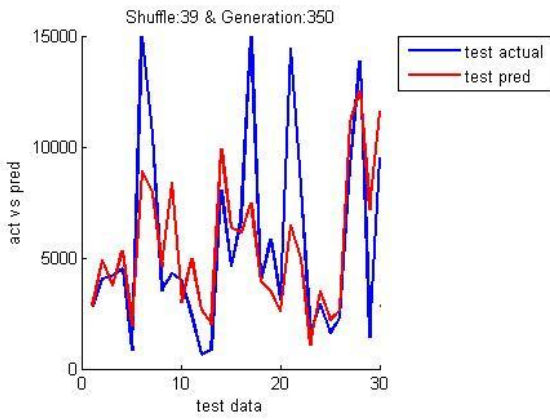
Εκτέλεση Β: Κόμβοι 16 - 73:



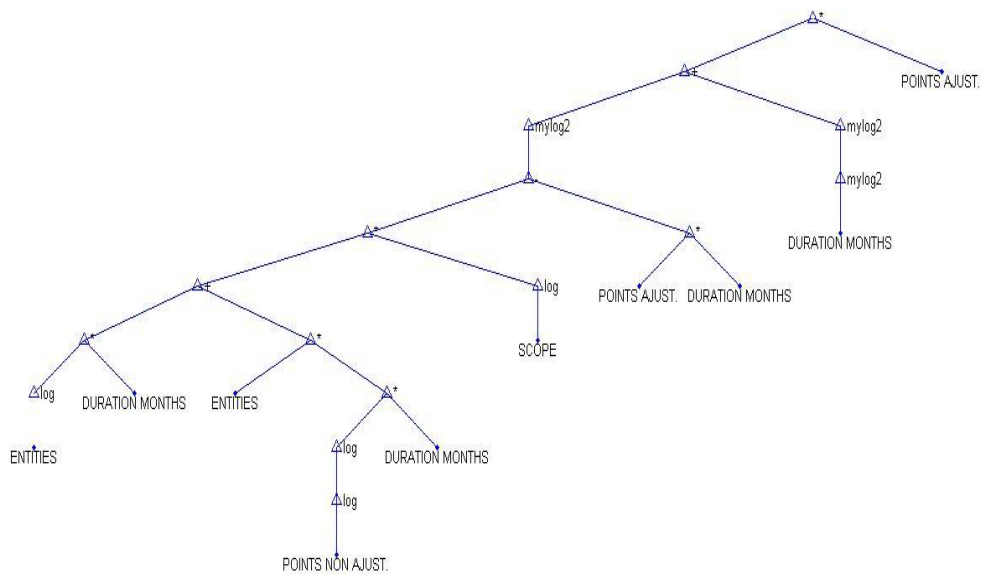
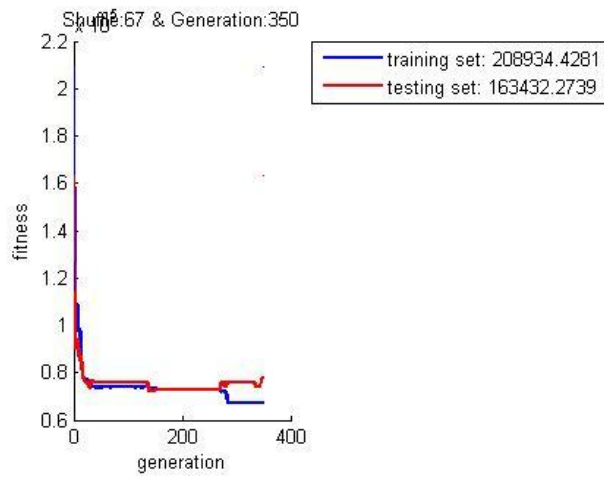
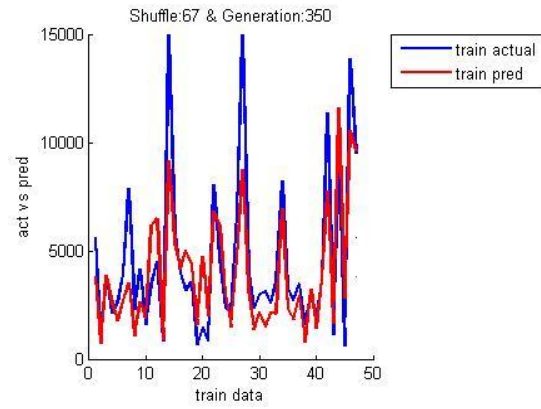
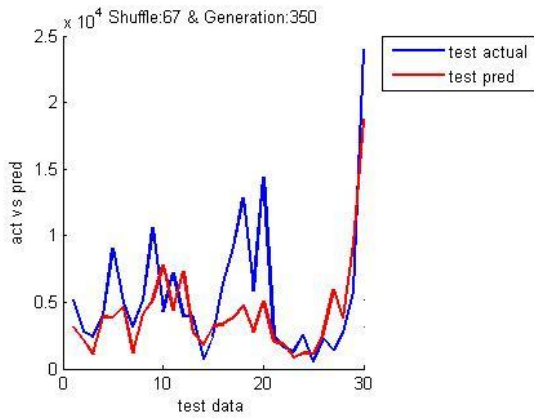
Εκτέλεση Β: Κόμβοι 20 - 4:



Εκτέλεση Β: Κόμβοι 28 - 39:

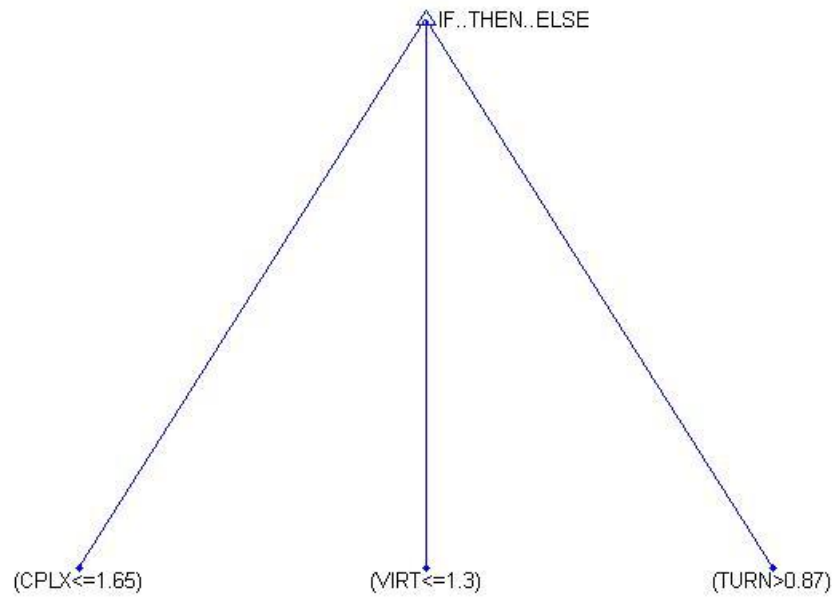


Εκτέλεση Β: Κόμβοι 28 - 67:

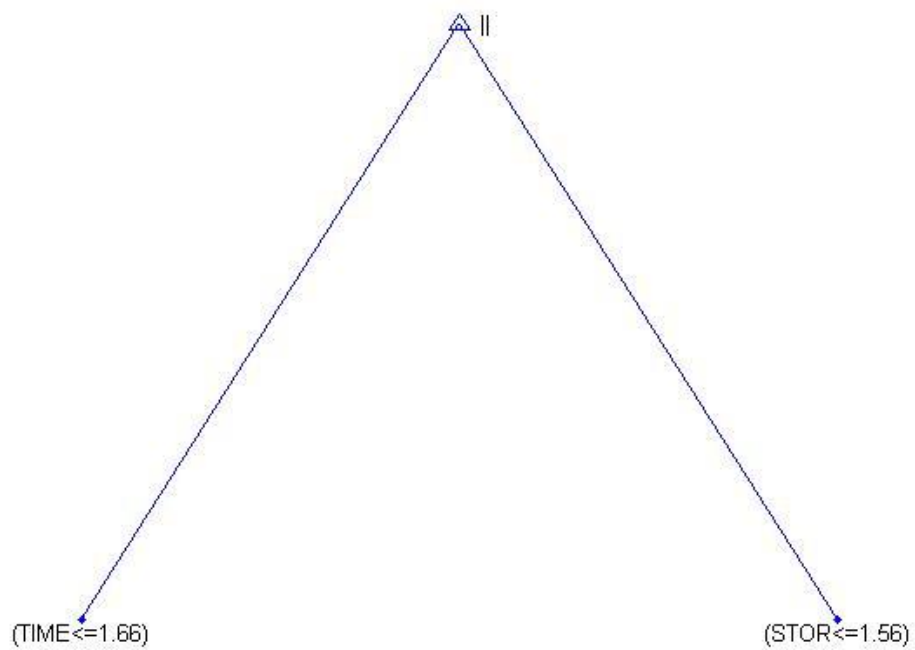


Β.2.3 Γραφικές Παραστάσεις Λογικών Εκτελέσεων - Cosmo

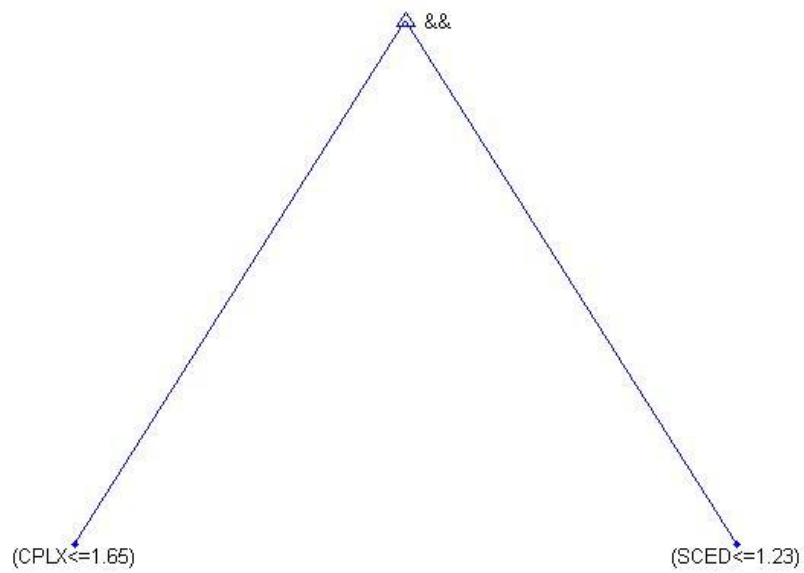
Εκτέλεση Μεγέθους 2 - 15:



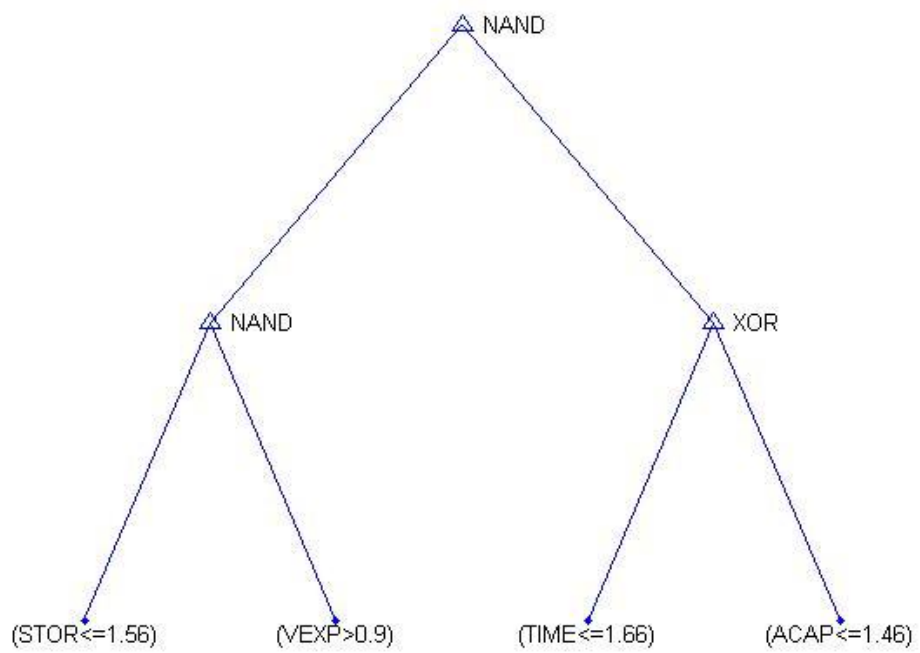
Εκτέλεση Μεγέθους 2 - 17:



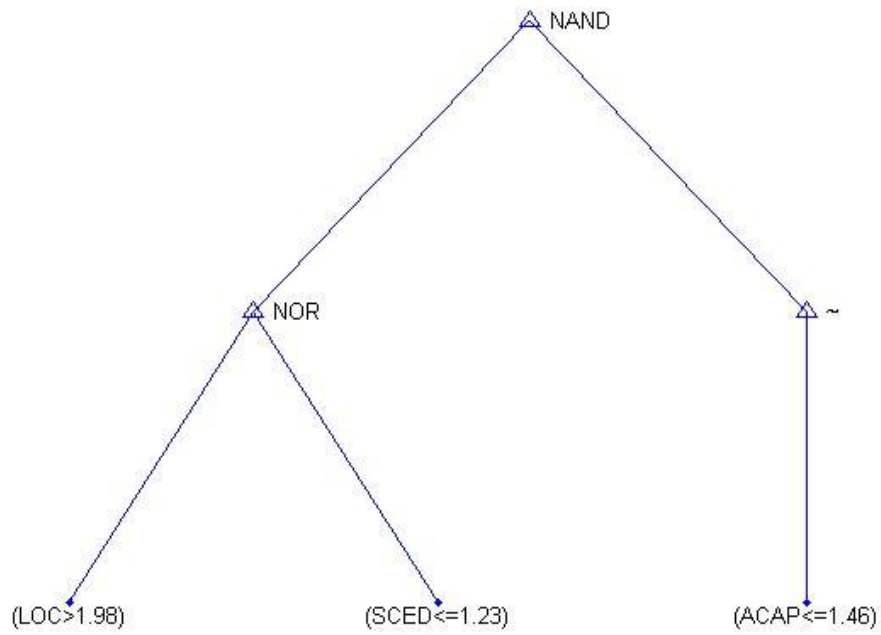
Εκτέλεση Μεγέθους 2 - 46:



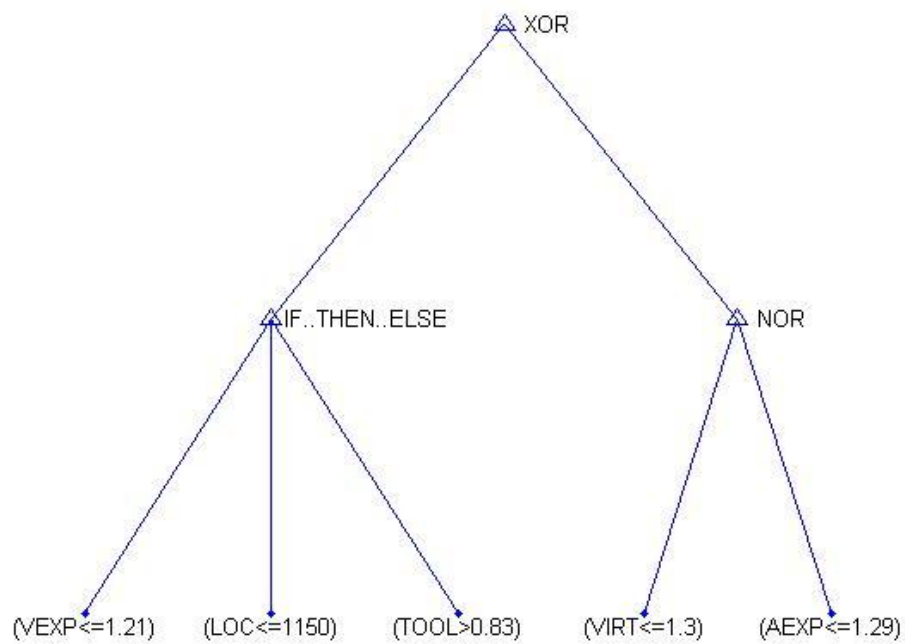
Εκτέλεση Μεγέθους 3 - 9:



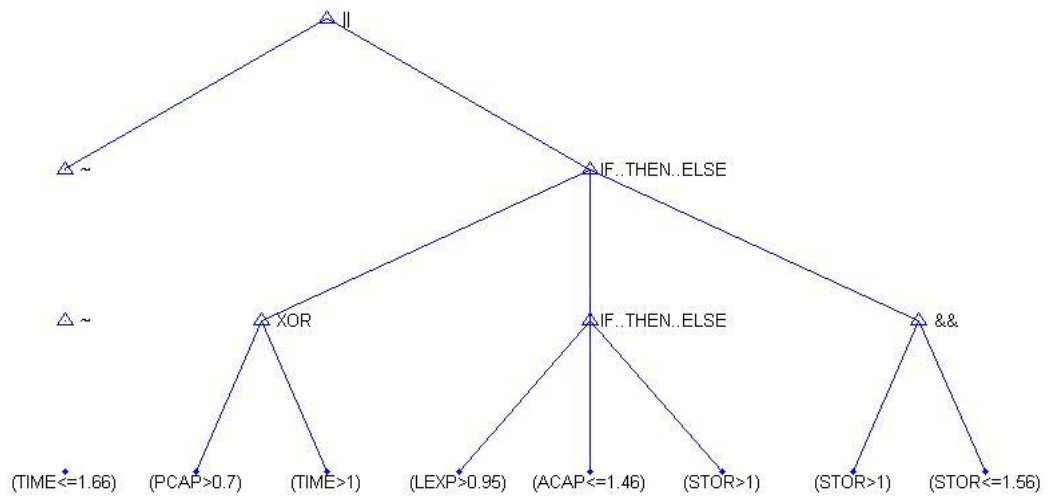
Εκτέλεση Μεγέθους 3 - 12:



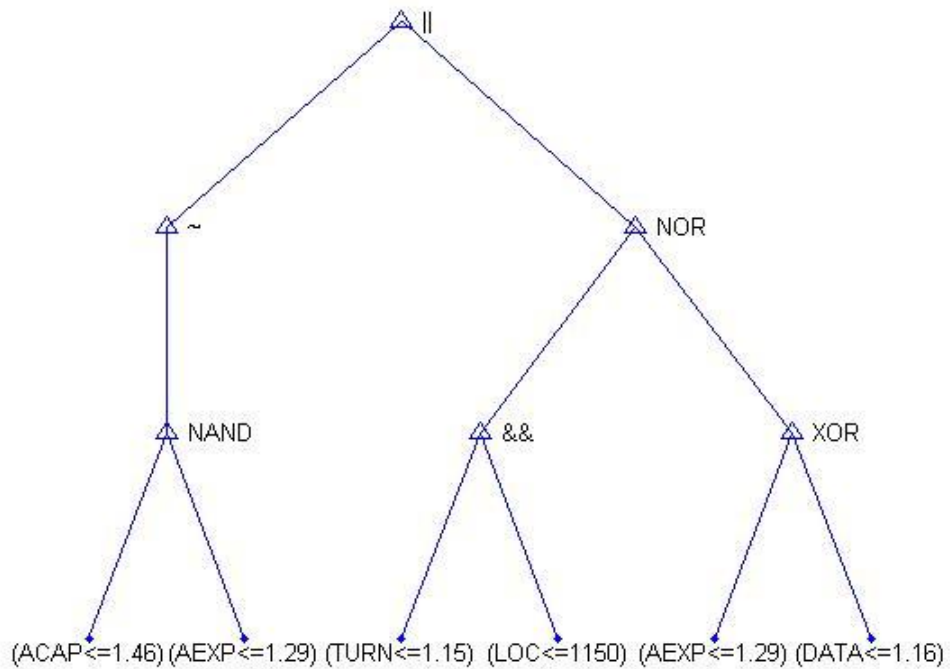
Εκτέλεση Μεγέθους 3 - 25:



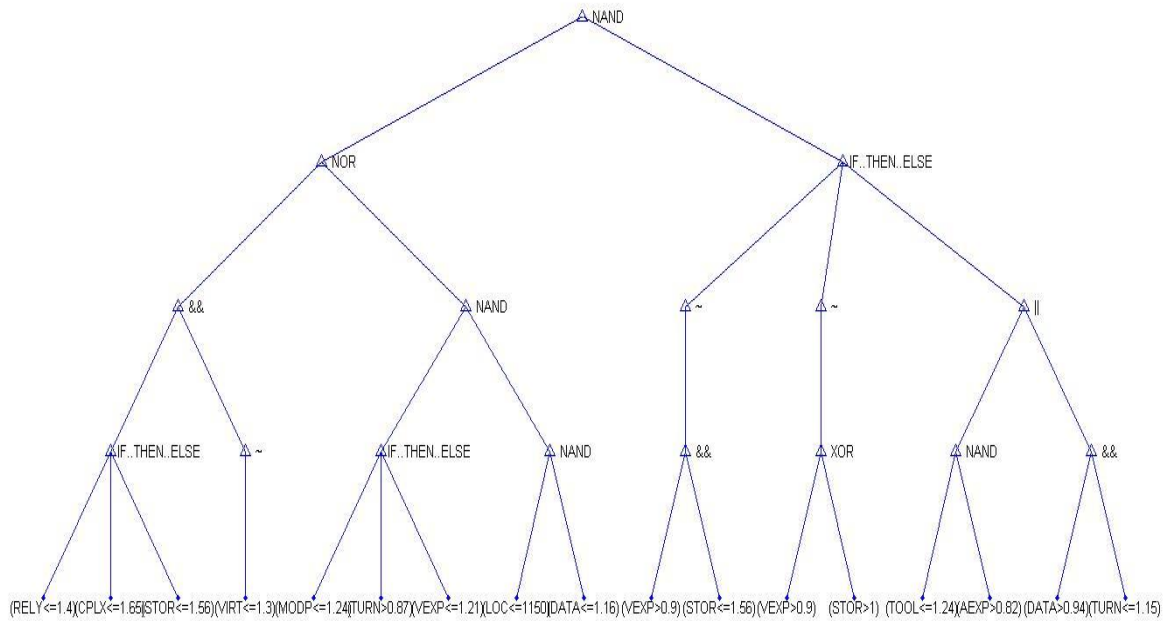
Εκτέλεση Μεγέθους 4 - 24:



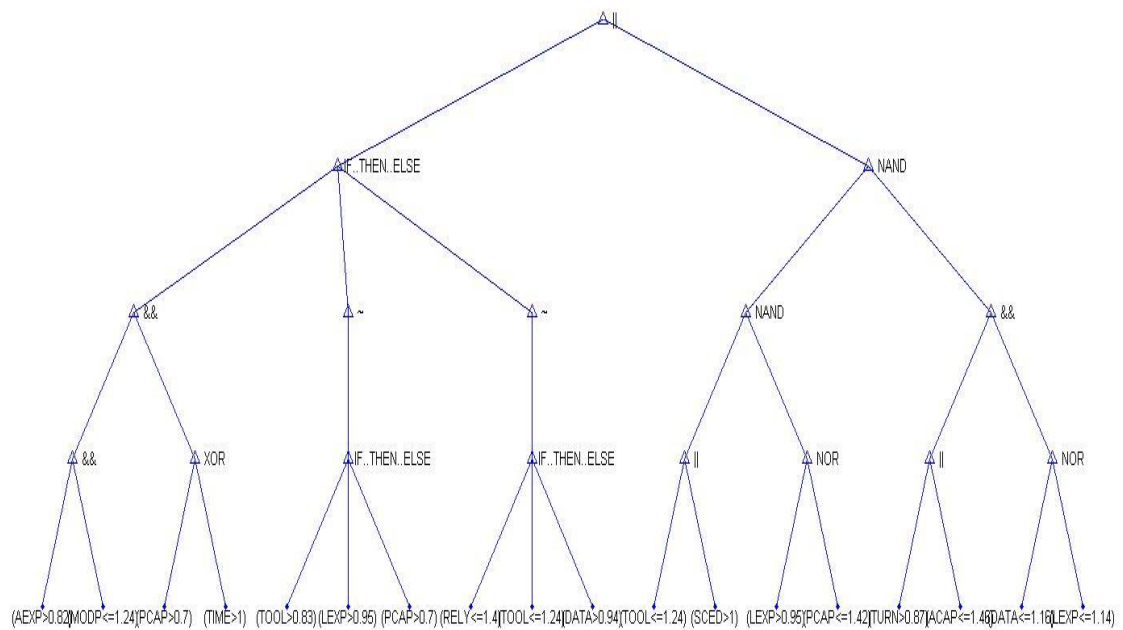
Εκτέλεση Μεγέθους 4 - 25:



Εκτέλεση Μεγέθους 5 - 21:

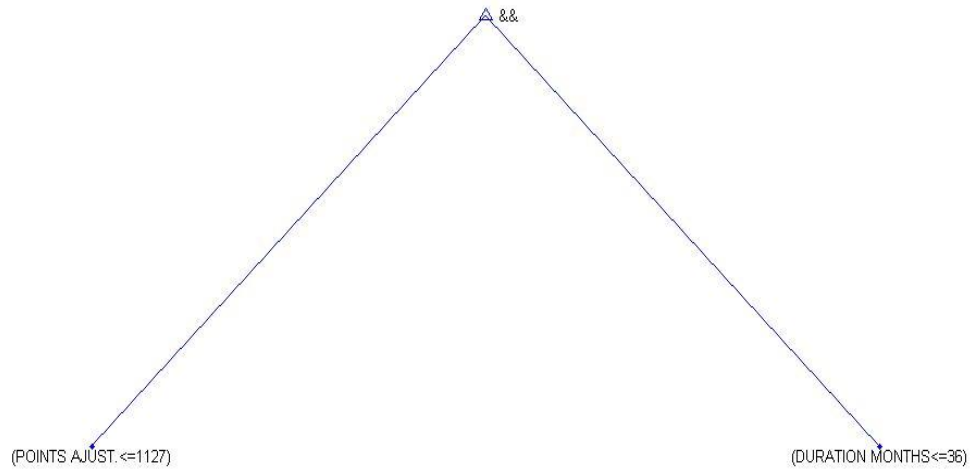


Εκτέλεση Μεγέθους 5 - 23:

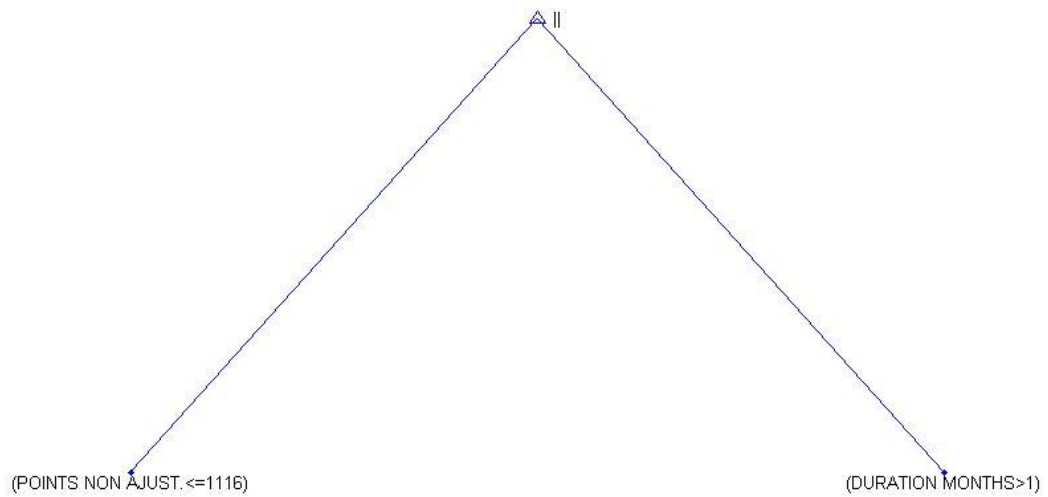


Β.2.4 Γραφικές Παραστάσεις Λογικών Εκτελέσεων - Desharnais

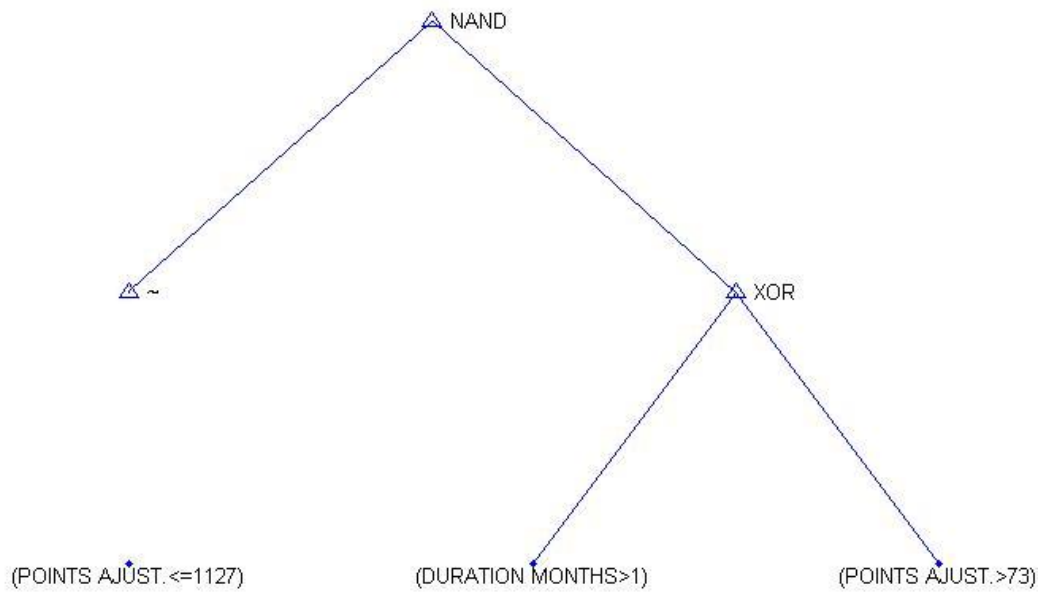
Εκτέλεση Μεγέθους 2 - 1:



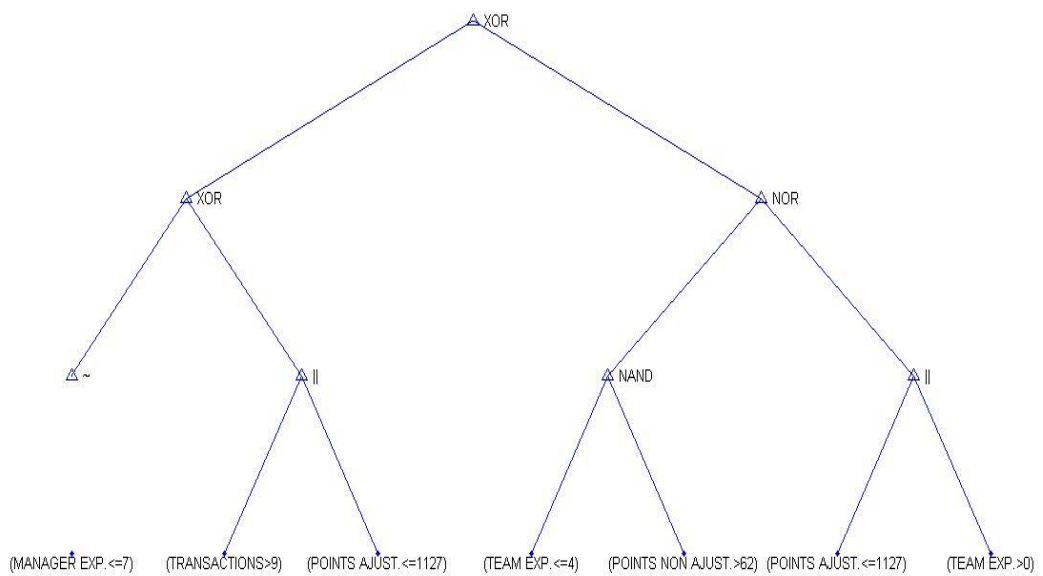
Εκτέλεση Μεγέθους 2 - 6:



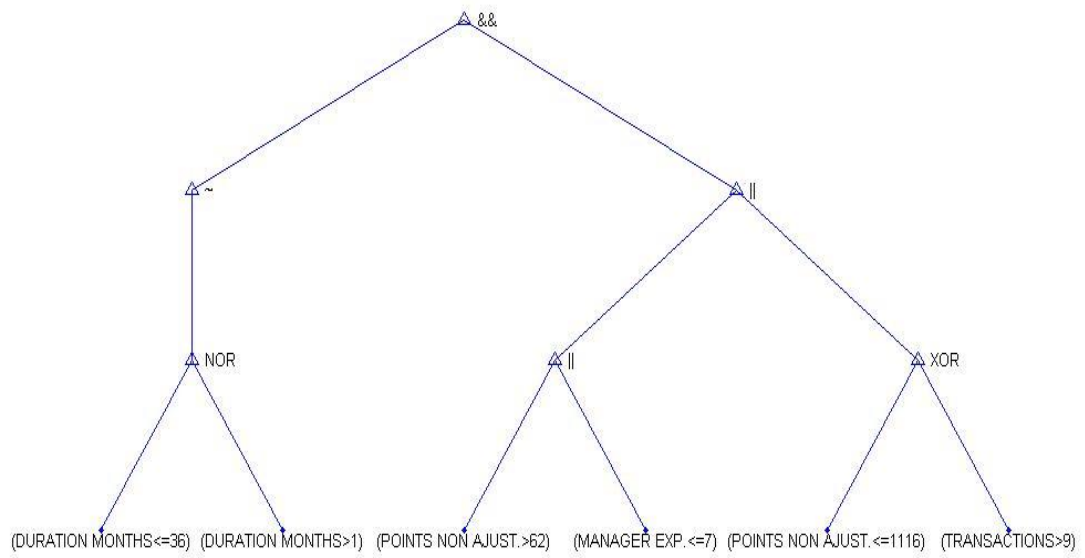
Εκτέλεση Μεγέθους 3 - 9:



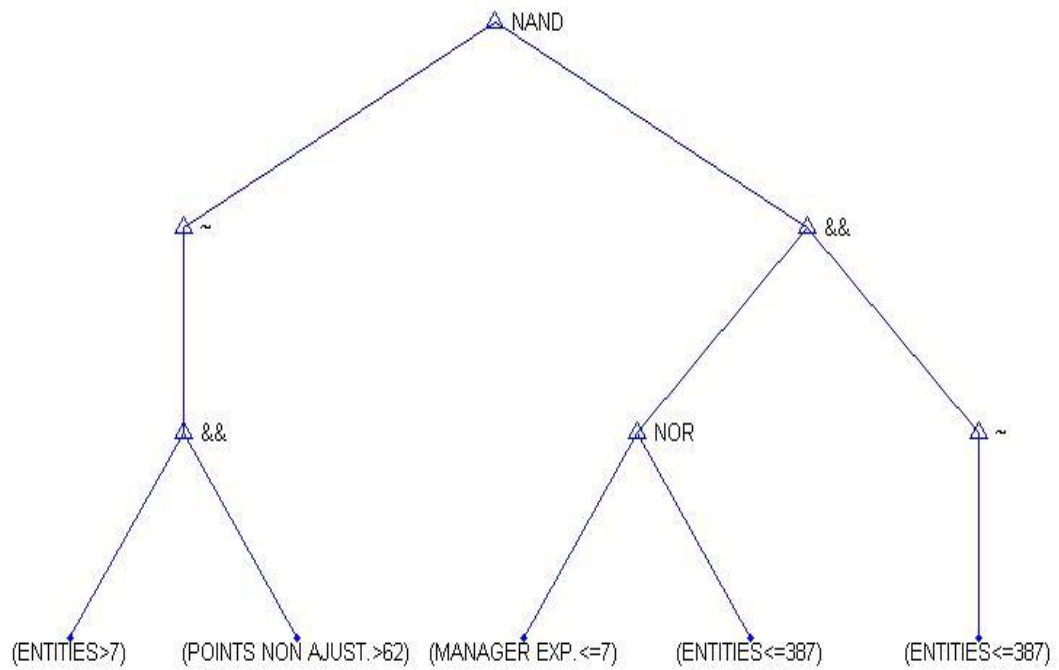
Εκτέλεση Μεγέθους 4 - 6:



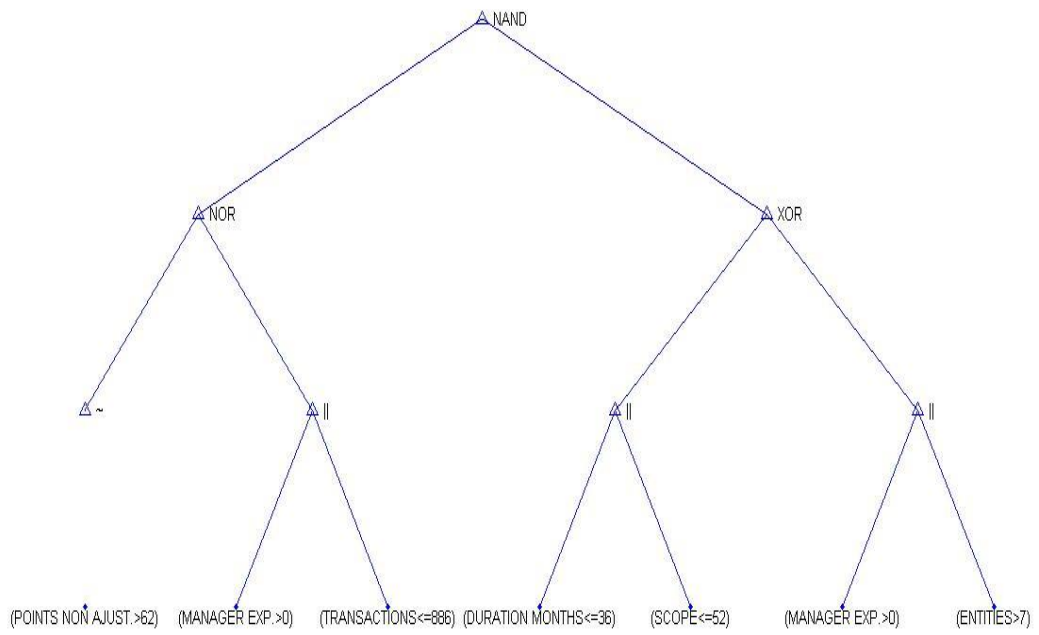
Εκτέλεση Μεγέθους 4 - 8:



Εκτέλεση Μεγέθους 4 - 27:

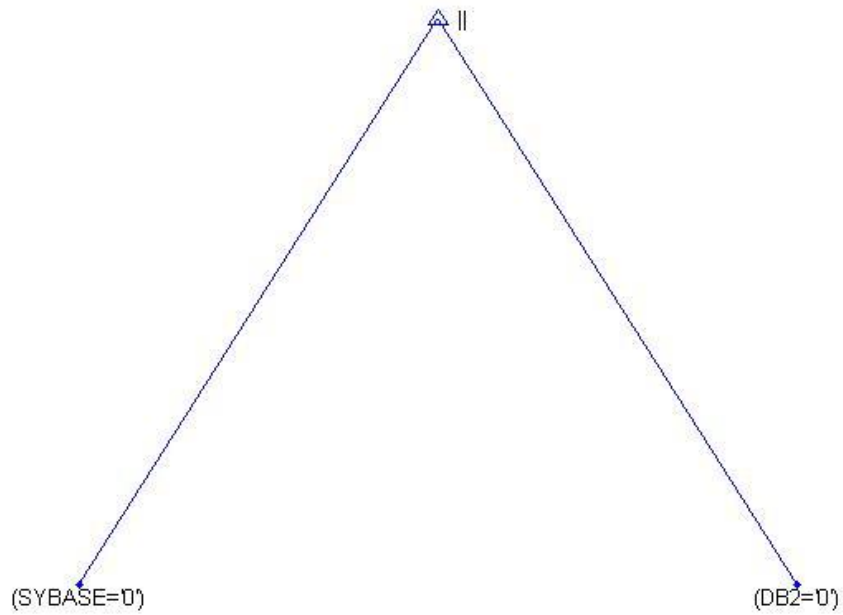


Εκτέλεση Μεγέθους 4 - 28:

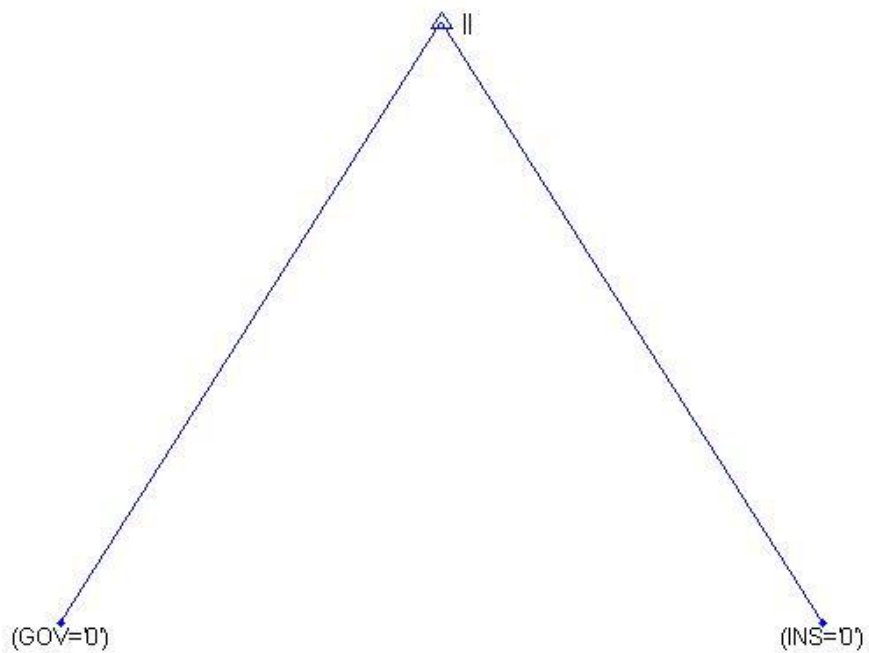


Β.2.5 Γραφικές Παραστάσεις Κατηγορικών Εκτελέσεων - ISBSG

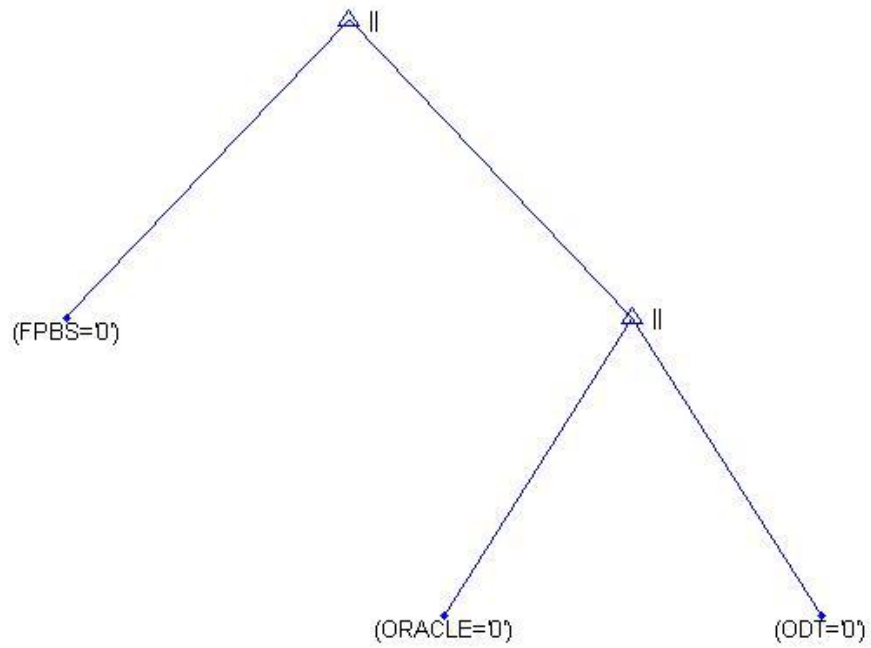
Εκτέλεση Μεγέθους 2 - 2:



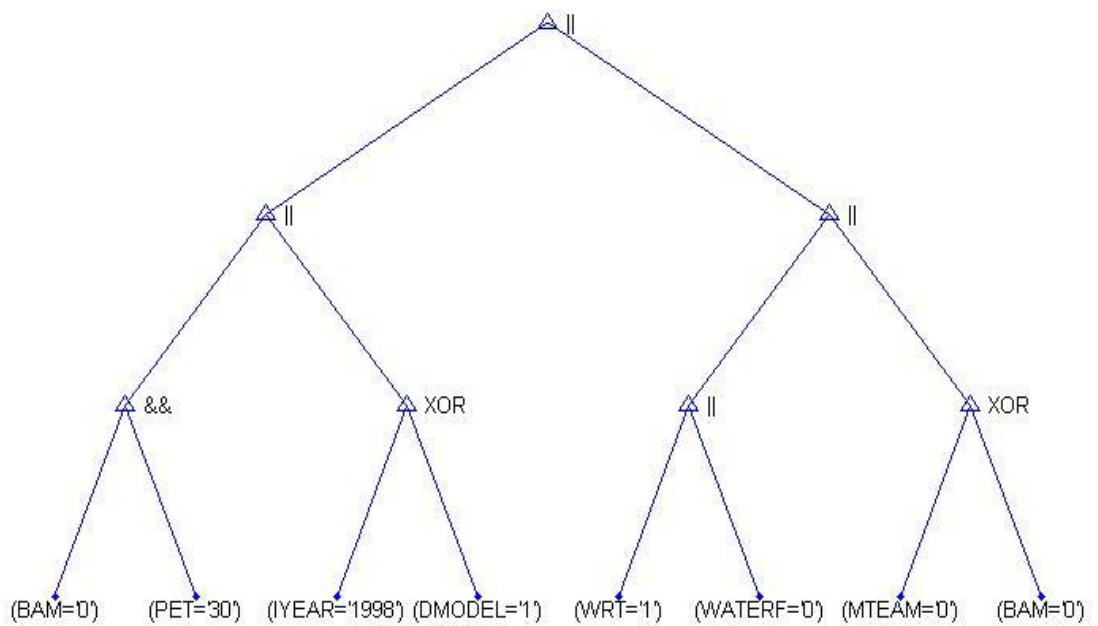
Εκτέλεση Μεγέθους 2 - 3:



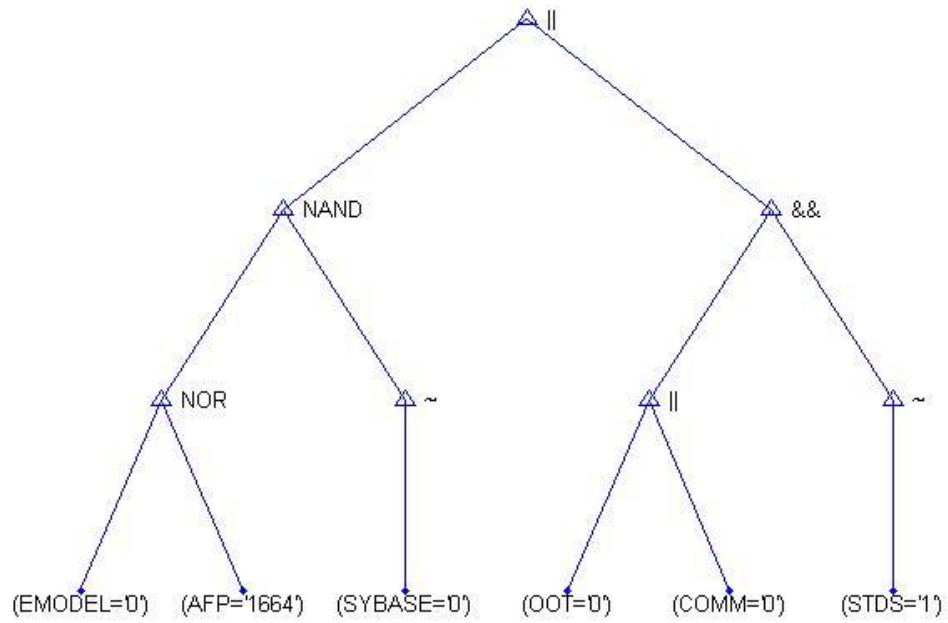
Εκτέλεση Μεγέθους 3 - 11:



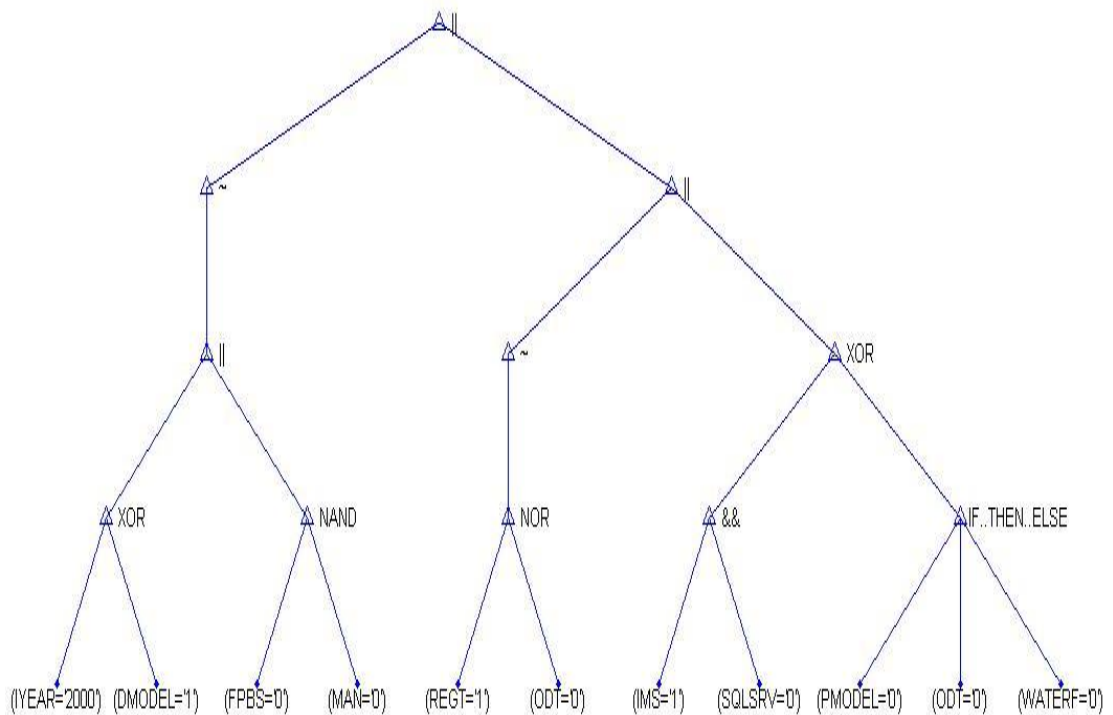
Εκτέλεση Μεγέθους 4 - 2:



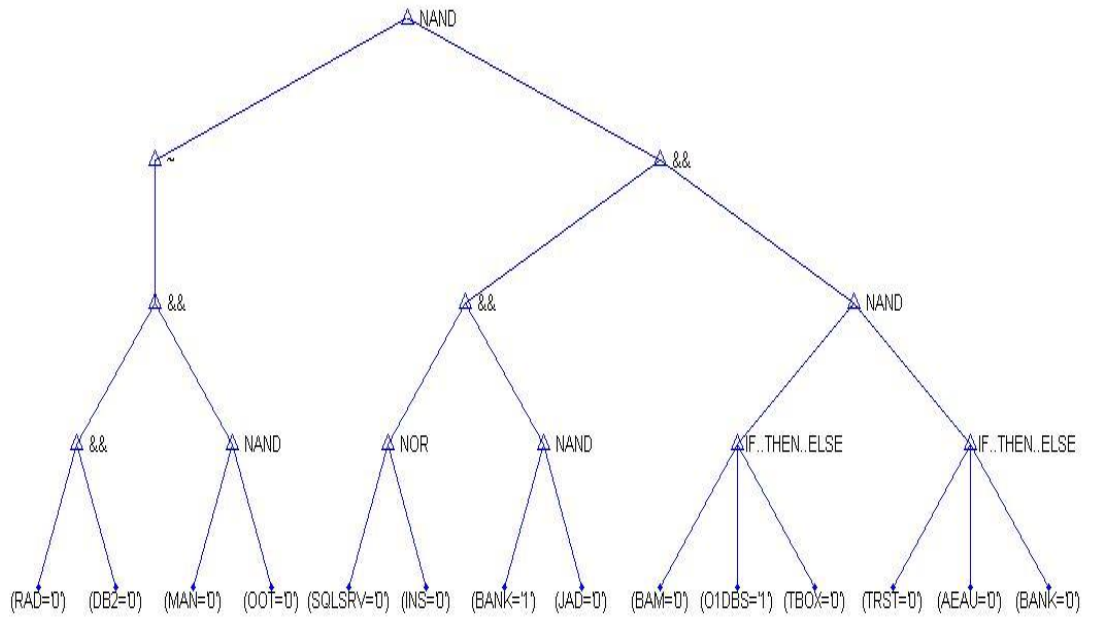
Εκτέλεση Μεγέθους 4 - 6:



Εκτέλεση Μεγέθους 5 - 2:



Εκτέλεση Μεγέθους 5 - 5:



Εκτέλεση Μεγέθους 5 - 7:

